



FEUP Universidade do Porto
Faculdade de Engenharia

Mestrado em Multimédia

Dissertação para a obtenção do grau de Mestre em
Multimédia

Sleepwalker

Um jogo feito em casa para
dispositivos móveis

Autor:

Hugo Miguel Gonçalves Crespo Machado da Silva

Orientador:

Professor Doutor Eurico Carrapatoso

Resumo

O mundo em que vivemos está em constante mudança. Nas últimas décadas a evolução das tecnologias analógicas para digitais tem vindo a disponibilizar novos meios de criação de conteúdos e a permitir o seu desenvolvimento a toda a população. O fruto desta liberdade de criação tem resultado em fotografias, desenhos, músicas ou vídeos que inundam a internet com níveis de qualidade variados, mas apesar da multiplicidade de ferramentas e meios a quantidade de jogos amadores continua parca em relação a outros tipos de media.

Esta dissertação foca-se na criação de jogos por pequenas equipas amadoras e pretende analisar os motivos que resultam nesta escassez. Devido à crescente fatia do mercado abrangida pelos dispositivos móveis e à simplicidade relativa dos jogos para este tipo de equipamentos, o estudo centra-se sobre os jogos destinados a esta gama de aparelhos.

A criação de um jogo é realizada desde a sua idealização até ao final do período de testes, utilizando ferramentas de fácil utilização que permitem o desenvolvimento do jogo por amadores sem grandes conhecimentos técnicos nas áreas de programação ou estruturas de dados. A ferramenta Game-Editor é utilizada para permitir uma experiência similar à necessária para a construção de um jogo por aficionados da informática na óptica do utilizador, ou aqueles que regularmente disponibilizam pequenas apresentações de PowerPoint ou Flash, vídeos de telemóvel ou músicas amadoras. A utilização desta ferramenta possibilita a exportação do jogo em variados formatos, executáveis em Windows, Linux, Windows Mobile Classic ou Professional ou a consola GP2X.

Devido à imensidão de jogos actualmente disponível para várias plataformas, pretende-se para o conceito um alto nível de originalidade, tentando distanciar-se das múltiplas variações de jogos que ao longo dos tempos alcançaram o sucesso. Vídeos virais viajam rapidamente – ainda que criados com câmaras ou telemóveis simples e acessíveis – entre caixas de correio electrónico, muitas vezes com razões bastante superiores a megaproduções. Para que tal aconteça o conteúdo destes vídeos apresenta algo de diferente, divertido, chocante ou original. Para alcançar um efeito similar com um jogo, a originalidade e a introdução de conceitos de mecânica menos comuns nos jogos existentes representa uma das finalidades da ideia apresentada.

A experiência obtida com o desenvolvimento de um conceito original resultou na criação de parte de um jogo com a finalidade de analisar o motivo que leva à discrepância entre a quantidade de jogos criados em casa e a de outros tipos de conteúdos desenvolvidos nas mesmas condições.

Paralelamente às finalidades apresentadas para esta dissertação, a experiência obtida com o estudo e o trabalho efectuado contribuiu para uma aprendizagem e consolidação de conhecimentos nas áreas de idealização, concepção e desenvolvimento de videojogos.

Abstract

The world we live in is constantly changing. In recent decades the evolution of analogue technologies to digital has been providing new ways of media content creation and allowing its development to the entire population. The fruit of this creative freedom has resulted in photographs, drawings, music or videos flooding the Internet with varying levels of quality, but despite the plethora of tools and means, the amount of amateur games is still meagre compared to other types of media.

This dissertation focuses on the creation of games by small amateur teams and aims to analyse the reasons that result in this shortage. Due to the increasing share of market covered by mobile devices and the relative simplicity of games for this type of equipment, the study centres on games for this kind of appliances.

The creation of a game is followed through from its conception to the end of the test period, using user-friendly tools that allow the development of the game by amateurs without technical expertise in the areas of programming or data structures. The Game-Editor tool is used to provide an experience similar to that required to build a game by IT fans in a user's perspective, or those who regularly create small PowerPoint or Flash presentations, amateur mobile phone videos or music. The use of this tool allows the user to export the game in various formats which can be ran on Windows, Linux, Windows Mobile Classic or Professional or the GP2X console.

Due to the availability of a wide range of games for multiple platforms, the achievement of a high level of originality is intended for the concept, trying to distance itself from the many variations of games that over the years achieved success. Viral videos travel quickly - even if created with basic and affordable cameras or mobile phones – between mailboxes, often faster and further than big productions. For this to happen, the content of the videos usually offers something different, fun, shocking or original. To achieve a similar effect with a game, the originality and the introduction of concepts of mechanics less common in the existing games is one of the purposes of the idea presented.

The experience gained from the development of an original concept resulted in the creation of part of a game in order to analyse the motives for the discrepancy between the amount of games created at home and other types of content developed under the same conditions.

In addition to the purposes described for this thesis, the experience gained from the study and work carried out contributed to the learning and consolidation of knowledge in the areas of ideation, design and development of videogames.

Agradecimentos

Este projecto não teria sido possível sem a ajuda e o apoio da minha família, dos meus professores e dos meus amigos. Serve esta página para transmitir os meus agradecimentos

Ao Professor Doutor Eurico Carrapatoso, meu orientador, pelo seu empenho e disponibilidade e pelo constante acompanhamento na realização desta dissertação.

Aos meus pais, os meus melhores amigos, aos meus avós e restante família que me apoiaram durante todo o meu percurso académico e profissional, e durante toda a minha vida, sem os quais este trabalho teria sido impossível.

Aos meus amigos e colegas de curso, em especial ao João Aires, João Sousa, Ana Quirino, Diogo Franco e Paula Carvalho, por todo o apoio e companheirismo.

Ao meu grande amigo João Miranda por estar sempre presente e compreender a minha ausência.

Ao Paulo Santana, pela ajuda especial na resolução de situações técnicas imprevistas.

Aos professores Carlos Guedes e José Azevedo, que se mostraram sempre motivadores, compreensivos e dispostos a ajudar.

A todas as pessoas que colaboraram no projecto com as suas opiniões e críticas e a todos aqueles que contribuíram para as análises de viabilidade e testes do protótipo.

À Joana Neto, um agradecimento muito especial pelo apoio constante desde que a conheci, por aceitar as minhas excentricidades sem nunca esperar nada em troca, pela ajuda na dissertação e na vida.

A todos, o meu mais sincero obrigado.

Índice de Tabelas

TABELA 1 – FERRAMENTAS DE PROTOTIPAGEM / DESENVOLVIMENTO.....	40
TABELA 2 – METODOLOGIA DE CRIAÇÃO DE UM JOGO.....	45
TABELA 3 – CRESCIMENTO DOS SISTEMAS OPERATIVOS PARA TELEMÓVEIS 2007-2008, EUA (GARTNER 2009)	65

Índice de Imagens

FIG. 1 – JOGO REAL DE UR – <i>BRITISH MUSEUM</i>	5
FIG. 2 – SENET, TABULEIRO ENCONTRADO NA TUMBA DE TUTANKHAMON	5
FIG. 3 – M&M'S MINI'S BLAST	7
FIG. 4 – VINÍCOLA, DE VITAL LACERDA – O PRIMEIRO JOGO PORTUGUÊS DE PROECÇÃO INTERNACIONAL. ...	7
FIG. 5 – CRAGSTAN PERISCOPE-FIRING RANGE	8
FIG. 6 – WACO'S TIC-TAC-TOE	8
FIG. 7 – MATTEL AUTO RACE E CAIXA	9
FIG. 8 – CONSOLA MICROVISION	10
FIG. 9 – MAGNAVOX ODYSSEY	11
FIG. 10 – ATARI PONG	12
FIG. 11 – ODYSSEY 100 E ODYSSEY 200	12
FIG. 12 – ATARI SUPER PONG	13
FIG. 13 – COLECO TELSTAR	13
FIG. 14 – ATARI 2600.....	14
FIG. 15 – ODYSSEY ² / PHILIPS VIDEOPAC G7000	15
FIG. 16 – MATTEL INTELLIVISION.....	15
FIG. 17 – VECTREX.....	16
FIG. 18 – COLECOVISION	16
FIG. 19 – NINTENDO ENTERTAINMENT SYSTEM	17
FIG. 20 – SEGA MASTER SYSTEM	18
FIG. 21 – SEGA MEGRA DRIVE	19
FIG. 22 – SNK NEO GEO COM CONTROLADOR.....	19
FIG. 23 – SUPER NINTENDO ENTERTAINMENT SYSTEM OU SNES	20
FIG. 24 – ATARI JAGUAR	20
FIG. 25 – SONY PLAYSTATION	21
FIG. 26 – NINTENDO 64	22
FIG. 27 – SEGA DREAMCAST.....	22
FIG. 28 – SONY PLAYSTATION 2.....	23
FIG. 29 – NINTENDO GAMECUBE	23
FIG. 30 – MICROSOFT XBOX	24
FIG. 31 – WIIMOTE E NINTENDO WII, SONY PLAYSTATION 3 E XBOX 360.....	25
FIG. 32 – NINTENDO GAME BOY	26
FIG. 33 – ATARI LYNX	26

FIG. 34 – SEGA GAME GEAR	27
FIG. 35 – GAME BOY ADVANCE ORIGINAL, SP E MICRO	28
FIG. 36 – NOKIA N-GAGE E N-GAGE QD	29
FIG. 37 – NINTENDO DSI BRANCA E PRETA.....	30
FIG. 38 – SONY PSP SIM & LITE E PSP GO	31
FIG. 39 – MULTITOUCH SPACE INVADERS NO FESTIVAL OFFF 09	32
FIG. 40 – TELEMÓVEL NOKIA N95.....	34
FIG. 41 – POCKETPC QTEK S100.....	35
FIG. 42 – APPLE IPHONE E T-MOBILE G1 (ANDROID)	35
FIG. 43 – GAMEPARK GP32	36
FIG. 44 – GAMEPARK HOLDINGS GP2X-F100	37
FIG. 45 – GAMEPARK HOLDINGS GP2X WIZ.....	38
FIG. 46 – DINGOO A-320	38
FIG. 47 – CONSOLA OPENPANDORA.....	39
FIG. 48 – TANDY TRS-80 IV	42
FIG. 49 – LITTLE BIG PLANET – 2008 <i>MEDIA MOLECULE, SCE/WWS</i>	43
FIG. 50 – STORYBOARD – HALO 2.....	46
FIG. 51 – ARTE CONCEPTUAL – FABLE 2.....	48
FIG. 52 – VISTA LATERAL	51
FIG. 53 – VISTA SUPERIOR.....	51
FIG. 54 – PERSPECTIVA ISOMÉTRICA	51
FIG. 55 – JOGO BASEADO EM “TILES”	52
FIG. 56 – SPRITE SHEET	53
FIG. 57 – POCKET SOKOBAN.....	54
FIG. 58 – AMBIENTE JOGÁVEL CRIADO COM <i>QUAKE ENGINE</i>	55
FIG. 59 – DIAGRAMA SIMPLIFICADO DA ARQUITECTURA DE JOGO	57
FIG. 60 – ARQUITECTURA DO CICLO PRINCIPAL DE JOGO (CSPROJECTS)	59
FIG. 61 – ESQUEMA DO PROCESSO DE DESENVOLVIMENTO E TESTES (THOMPSON, BERBANK-GREEN, AND CUSWORTH 2007)	60
FIG. 62 – PLANTA DE TRÊS LABORATÓRIOS DE TESTES NOS MICROSOFT GAME STUDIO.....	62
FIG. 63 – LABORATÓRIO DE TESTES DOS MICROSOFT GAMES STUDIO.....	62
FIG. 64 – TOP DE VENDAS DE JOGOS NA HANDANGO	66
FIG. 65 – BEST SELLERS NA POCKETGEAR	66
FIG. 66 – MAPA DE GANTT – PRAZOS DE DESENVOLVIMENTO.....	68
FIG. 67 – SLEEPWALKER - PROTÓTIPO INICIAL, VERSÃO 2	69

FIG. 68 – TOMB RAIDER – BOTÕES VIRTUAIS EM WINDOWS MOBILE	73
FIG. 69 – DESENHO INICIAL DO NÍVEL DA FLORESTA	74
FIG. 70 – RESULTADO FINAL DO NÍVEL DA FLORESTA	74
FIG. 71 – ARTE CONCEPTUAL - GODOFREDO	75
FIG. 72 – ARTE CONCEPTUAL – IDEIA ORIGINAL PARA O ADJUVANTE DE GODOFREDO	76
FIG. 73 – ARTE CONCEPTUAL – INÁCIO.....	76
FIG. 74 – ARTE CONCEPTUAL – CASA DO URSO GODOFREDO.....	77
FIG. 75 – GAME-EDITOR – CAIXA DE DIÁLOGO PARA CONTROLO DOS ACTORES.....	78
FIG. 76 – ANIMAÇÃO DA SETA QUE APONTA NA DIRECÇÃO DE GODOFREDO	79
FIG. 77 – GODOFREDO VECTORIZADO E PREENCHIDO A PARTIR DA ARTE CONCEPTUAL	81
FIG. 78 – ANIMAÇÃO FINAL DO MORCEGO	82
FIG. 79 – MOLA MODELADA NO PROGRAMA CINEMA4D	83
FIG. 80 – MELODIA E CONTRA MELODIA, SECONDARY RAG E CONTRABAIXO NO TEMA DA PRIMEIRA PARTE DA SECÇÃO A DA FLORESTA.	85
FIG. 81 – EFEITO “MOONWALK”	91

Lista de Acrónimos

AAC - *Advanced Audio Coding*
AES - *Advanced Entertainment System*
API - *Application Programming Interface*
ARM – *Advanced Risc Machine*
BLU – *Back-Lit Unit*
CD – *Compact Disk*
CEO – *Chief Executive Officer*
DVD - *Digital Versatile Disc ou Digital Video Disc*
FLU – *Front-Lit Unit*
FPS – *First Person Shooter*
GBA – *Game Boy Advance*
GBC – *Game Boy Color*
GDD – *Game Design Document*
GPS – *Global Positioning System*
GPU - *Graphics Processing Unit*
GSM – *Global System for Mobile communications*
HD – *High Definition*
HSDPA – *High-Speed Downlink Packet Access*
HTC – *High Tech Computer*
J2ME – *Java 2 platform, Micro Edition*
JVM – *Java Virtual Machine*
LCD – *Liquid Crystal Display*
LDD – *Level Design Document*
LED - *Light Emitting Diode*
MIDI - *Musical Instrument Digital Interface*
MMORPG - *Massively Multiplayer On-line Role Playing Game*
MP3 – *MPeg-1 audio layer 3*
MP4 – *MPeg-4*
MPEG – *Moving Picture Experts Group*
MVS - *Multi Video System*
NDS – *Nintendo Dual-Screen*
NEC - *Nippon Electric Company*
NES – *Nintendo Entertainment System*
NPC – *Non-Playing Character*
NURBS - *Non-Uniform Rational B-Spline*
OGRE - *Object-Oriented Graphics Rendering Engine*
PCM - *Pulse Code Modulation*
PDA – *Personal Digital Assistant*
PNG - *Portable Network Graphics*
PS2 – *Playstation 2*
PSP – *Play Station Portable*
QVGA – *Quarter Video Graphics Array*

RIM – *Research in Motion*
RISC – *Reduced Instruction Set Computer*
RPG – *Role Playing Game*
SD – *Secure Digital*
SDI - *Simple DirectMedia Layer*
SDK – *Software Development Kit*
SNES – *Super Nintendo Entertainment System*
SNK - *Shin Nihon Kikaku (New Japan Project)*
TFT – *Thin-Film Transistor*
TRS-80 – *Tandy Radio Shack-80*
TSR - *Tactical Studies Rules, Inc.*
UMD – *Universal Media Disk*
UMTS – *Universal Mobile Telecommunications System*
USB – *Universal Serial Bus*
VCS – *Video Computer System*
VGA – *Video Graphics Array*
WVGA – *Wide Video Graphics Array*
WYSIWYG - *What You See is What You Get*
XGP – *eXtreme Game Player*

Índice

RESUMO	I
ABSTRACT	II
AGRADECIMENTOS	III
ÍNDICE DE TABELAS	IV
ÍNDICE DE IMAGENS	V
LISTA DE ACRÓNIMOS.....	VIII
1. INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 OBJECTIVOS	2
1.3 ESTRUTURA.....	2
2. HISTÓRIA DOS JOGOS	4
2.1 PRIMEIROS JOGOS	4
2.2 JOGOS DE TABULEIRO.....	5
2.3 JOGOS ELECTRÓNICOS	7
2.4 VIDEOJOGOS	11
2.5 CONSOLAS DE JOGOS	17
2.5.1 Consolas de Sala	17
2.5.2 Consolas Portáteis.....	25
3. ESTADO DA ARTE.....	33
3.1 EQUIPAMENTOS.....	33
3.2 FERRAMENTAS DE DESENVOLVIMENTO E DE PROTOTIPAGEM	39
3.3 JOGOS.....	42
4. DESENHO DE UM JOGO	45
4.1 PLANEAMENTO	46
4.2 DOCUMENTOS DE APOIO	48
4.2.1 Game Design Document	48
4.2.2 Level Design Document.....	50
4.3 CRIAÇÃO DE CONTEÚDOS	50
4.4 PROGRAMAÇÃO	54
4.5 TESTES	61
5. PROJECTO.....	64
5.1 ANÁLISE DE VIABILIDADE	64
5.2 PLANEAMENTO	67
5.3 FERRAMENTAS UTILIZADAS	70
5.4 ESPECIFICAÇÕES PARA FUTURO	71
6. IMPLEMENTAÇÃO	74

6.1	IDEALIZAÇÃO	74
6.2	ARTE	75
6.3	PROGRAMAÇÃO.....	78
6.4	SPRITES E ANIMAÇÃO	81
6.5	SOM.....	84
6.5.1	Música.....	84
6.5.2	Efeitos Sonoros.....	85
7.	AVALIAÇÃO	87
7.1	ACEITAÇÃO	87
7.2	TESTES.....	89
7.2.1	Usabilidade	89
7.2.2	“Bugs” e erros de desenvolvimento	91
8.	CONCLUSÃO	93
	REFERÊNCIAS BIBLIOGRÁFICAS	96

1. Introdução

Gradualmente tem vindo a ser concedida ao consumidor de media a possibilidade de produção dos seus próprios conteúdos. Se as ferramentas de criação de ilustrações, pinturas ou textos escritos desde há muito são disponibilizadas ao utilizador comum, outros media mais recentes, como a fotografia ou o vídeo foram em tempos reservados a pessoas com formação especializada.

Com a massificação das câmaras fotográficas, especialmente após a generalização da fotografia digital, também a criação deste tipo de conteúdos foi permitida às massas. O mesmo aconteceu com o vídeo, e tanto este como a fotografia podem ser produzidos com dispositivos que usamos no dia-a-dia, como o telemóvel.

Os jogos, por seu lado, continuam a ser realizados quase sempre por empresas, e apenas em raras situações por pequenas equipas ou programadores individuais. Cada vez mais circulam apresentações informáticas com desenhos, fotografias, textos, vídeos ou música produzidos por pessoas sem formação específica, mas jogos continuam a ser maioritariamente provenientes das mesmas fontes de há dez anos.

Esta dissertação pretende analisar técnicas e ferramentas, sempre que possível gratuitas, que permitem a qualquer pessoa criar um jogo. Os jogos estudados centralizam-se em plataformas móveis pela sua portabilidade e pela simplicidade deste tipo de jogo em relação aos que correm em consolas de sala da Sony, Microsoft ou Nintendo. Devido à dificuldade e custos de programação, execução, licenciamento e distribuição, as consolas portáteis Sony PSP ou Nintendo DS ficam também fora do contexto desta exposição.

1.1 Motivação

O ano de 1982 trouxe o nascimento de um dos mais carismáticos computadores da história da informática. Muitos foram os jovens que se iniciaram na programação com o intuito de alterar ou mesmo criar os seus próprios jogos para o ZX Spectrum. Este texto é motivado por uma enorme paixão que surgiu no início da década de 80 e que continua a aumentar.

Em vinte e sete anos, o número de jogos aumentou exponencialmente, havendo neste momento milhares de jogos que cobrem as mais diversas áreas. Aumentou também a capacidade dos computadores, apareceram processadores dedicados aos gráficos e à física, assim como novos dispositivos de controlo. A maioria dos jogos para computador ou consola de sala são quase fotorrealistas ou pelo menos graficamente muito exigentes, e os jogos mais parecidos com os da altura correm agora em telemóveis e dispositivos portáteis. Não sendo os mais apelativos visualmente, jogos deste tipo dependem muito dos princípios necessários para cativar o jogador pela “jogabilidade”. Criar um bom jogo exige vários saberes, cada um dos quais representa um desafio em si, tornando o desenvolvimento de videojogos uma actividade extremamente recompensadora.

O mercado dos jogos para dispositivos móveis tem um imenso potencial. Os equipamentos são cada vez mais numerosos e poucas são as pessoas que não andam constantemente acompanhadas de um telemóvel. Desde jogos didácticos a simples *puzzles* que nos mantêm ocupados nos momentos mortos, as aplicações lúdicas de bolso têm um enorme mercado (Hashim, Ab Hamed, and Sabri 2007).

A motivação para este documento deriva do afecto a certos tipos de jogos que marcaram uma época, do desafio que representa a recriação e inovação desse tipo de jogos, da multidisciplinaridade do tema e do mercado existente para a sua distribuição.

1.2 Objectivos

O intuito deste estudo é a criação de um jogo usando, sempre que possível, ferramentas de utilização simples, ao alcance de qualquer pessoa. Como objectivo da tese, é proposta uma pergunta que se espera ver resolvida no final do trabalho: “Será possível a criação de um jogo por um aficionado como é possível a criação de elementos individuais da matriz dos media?”.

De tempos em tempos surge um jogo comentado no mundo inteiro, cativando enormes números de adeptos. Jogos como Tetris, Lemmings ou Super Mario são conhecidos internacionalmente, não pelos incríveis gráficos ou som real 5.1, mas por conseguirem que o jogador volte a jogar vezes sem conta, sem perder o interesse. Este efeito é tido como um objectivo secundário deste trabalho, sendo que o jogo proposto deverá ser cativante, original, um jogo que cause os efeitos referidos, utilizando as mesmas ou outras técnicas.

Após o desenho do jogo e da sua mecânica, é proposto o desenvolvimento de um protótipo funcional, um nível do jogo que contenha os elementos que o tornam diferente. O desenvolvimento de pequenos protótipos fará parte da tarefa.

Um último objectivo do desenvolvimento desta tese é o estudo e a consequente aprendizagem e consolidação de conhecimentos, servindo este trabalho também para aprimorar as capacidades de criação e desenvolvimento de jogos.

1.3 Estrutura

Esta dissertação está organizada em oito capítulos. Iniciando-se com a exposição da motivação que levou ao seu desenvolvimento e os objectivos a atingir no final do trabalho, a introdução do documento termina com esta descrição da estrutura definida.

Os jogos digitais e as consolas de jogos, que nos dias de hoje são conhecidos e considerados banais pela maioria da população mundial resultam do aprimorar de ideias lúdicas combinadas com os avanços da tecnologia. No capítulo 2 uma apresentação histórica inicia-se com os primeiros jogos conhecidos da humanidade e termina nas gerações mais recentes de consolas de jogos. Por representarem não só uma alternativa mas também um antecedente histórico das consolas portáteis, os jogos electrónicos e as consolas de sala são apresentadas desde a sua infância até aos tempos actuais.

O capítulo 3, relativo ao estado da arte, está dividido em três secções consideradas relevantes para a criação de um jogo. Por um lado são estudados os mais actuais equipamentos, visto ser necessária ao criador de jogos ter a consciência do poder de processamento, portabilidade, autonomia, ergonomia e demais características das plataformas alvo do seu trabalho. É-lhe também importante a noção daquilo que o futuro reserva, de forma a permitir o desenvolvimento de ideias para novos jogos, continuações de jogos criados ou o desenvolvimento de novas funcionalidades, níveis ou *add-ons*.

As últimas novidades em termos de ferramentas de desenvolvimento e prototipagem são também expostas neste terceiro capítulo. A utilização de ferramentas actuais é cada vez mais importante no desenvolvimento de todo o tipo de aplicações, das quais os jogos são exemplos. Sem estes meios é praticamente impossível obedecer aos prazos curtos de desenvolvimento necessários para responder às exigências do mercado actual. Sem a ferramenta de prototipagem rápida Game-Editor, ou similar, seria muito difícil completar o projecto em tempo útil.

Para finalizar a informação relativa ao estado da arte é feita uma breve análise dos jogos mais actuais. Os jogadores estão cada vez mais exigentes e sem a adição de estratégias originais poucos seriam neste momento os adeptos dos jogos de outrora.

O quarto capítulo, dedicado ao estudo do desenho de um jogo, expõe as principais fases e documentos relativos à criação de jogos, focando a atenção numa metodologia de criação de um jogo por uma pequena equipa, no presente caso constituída por apenas um elemento. É descrita a importante etapa de planeamento, os dois mais importantes documentos utilizados no processo, a criação de conteúdos e a implementação do jogo, terminando com um breve estudo da importante fase de testes.

Uma análise de viabilidade baseada num estudo do mercado de telemóveis com Windows Mobile, um inquérito desenvolvido para estudar os hábitos de jogo em Portugal e uma revisão dos lugares cimeiros das tabelas de vendas de *software* para Windows Mobile iniciam o capítulo 5 indicando a viabilidade do projecto desenvolvido. O planeamento do trabalho e a descrição das ferramentas utilizadas continuam este capítulo antecedente ao da criação do protótipo. Para terminar a secção relativa ao projecto são apresentadas algumas ideias de especificações futuras que permitiriam expandir o jogo ou o leque de terminais em que este poderá ser executado.

A implementação de um protótipo funcional é descrita no capítulo 6, começando pelas ideias que deram origem ao projecto e continuando com as várias áreas de diferentes competências que entram no desenvolvimento de um trabalho deste tipo.

Com o protótipo completo, o penúltimo capítulo comunica uma série de testes internos e externos – desenvolvidos pelo criador do jogo e por pessoas alheias ao Projecto – por meio de um inquérito respondido após experimentar o jogo. Estes testes combinados com a exposição de ideias relativas ao projecto avaliam o desenrolar do trabalho e o produto final.

Finalmente são apresentadas sugestões para trabalho futuro e expostas as conclusões obtidas durante o decorrer e após o final do projecto.

2. História dos jogos

‘Play is a voluntary activity or occupation executed within certain fixed limits of time and place, according to rules freely accepted but absolutely binding, having its aim in itself and accompanied by a feeling of tension, joy and the consciousness that it is “different” from “ordinary life”.’ – (Huizinga 1970)

Os jogos são actividades que surgem naturalmente entre elementos da espécie humana. Desde a nossa infância que inatamente utilizamos actividades recreativas tão simples como a do bebé que esconde a cara entre os cobertores para surgir novamente com um sorriso, como que num “jogo das escondidas” de regras simples. O mesmo acontece entre animais, sendo comuns as brincadeiras entre os mais jovens elementos de uma ninhada. Embora não existam registos, jogos simples terão sido postos em prática desde antes do início da humanidade.

Os jogos actuais são o resultado de uma evolução e não uma arte nova. Elementos como o factor sorte dos mais antigos jogos de tabuleiro continuam presentes em muitos jogos. Na maior parte das situações existe um herói que deve recolher objectos – desde frutas, estrelas ou anéis, em jogos de plataformas, até armas ou escudos em FPSs, tal como nos primeiros jogos portáteis *Game & Watch*. Cada elemento original introduzido num jogo ao longo da história faz agora parte de uma paleta de artefactos que pode ser utilizada num jogo criado de raiz.

2.1 Primeiros Jogos

São conhecidas evidências da existência de jogos desde os primeiros registos históricos. Um dos primeiros jogos existentes de que se tem conhecimento é o “Senat”, mais conhecido na sua forma internacional como “Senet”. As primeiras provas da existência deste jogo datam da era de XXXV a.C., tendo mais de quarenta tabuleiros de jogo sido descobertos (Ali 2007), alguns em excelentes condições. Um tabuleiro completo com as peças do jogo foi encontrado no túmulo do faraó Tutankhamon. Outro dos jogos mais antigos conhecidos, também jogado pelos egípcios é o “Jogo Real de Ur”, datado do século XXVI a.C. Ambos os jogos utilizavam já elementos usados nos jogos actuais. O Jogo de Ur era constituído por catorze peças, sendo sete de um lado e sete do outro, e três dados em forma de pirâmide (Soubeyrand 1996). O elemento aleatório introduzido pelos dados é uma contribuição valiosa para a jogabilidade e continua a estar presente em grande parte dos jogos da actualidade, passados mais de cinco milénios.

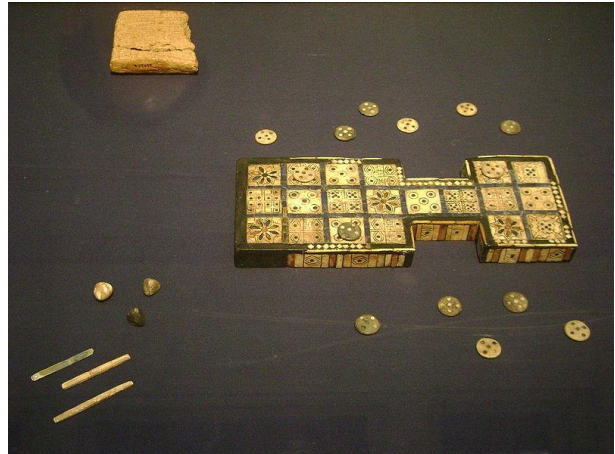


Fig. 1 – Jogo Real de UR – *British Museum*



Fig. 2 – Senet, tabuleiro encontrado na tumba de Tutankhamon

O jogo “Go” tem mais de quatro mil anos de existência (Chikun 1997) e é um dos precursores do Xadrez, este último com cerca de mil anos de existência. Ambos partilham componentes simples – múltiplas peças em duas cores, criando no entanto múltiplas estratégias extremamente complexas.

2.2 Jogos de Tabuleiro

Com o passar do tempo, e apesar da evolução da tecnologia, os jogos de tabuleiro continuam tão ou mais jogados como antes, tendo novos conceitos sido introduzidos e alguns mantidos ou melhorados.

Noções importantes como estratégia ou planeamento são utilizadas nos mais variados jogos, e essenciais para ganhar ou mesmo completar o jogo.

O jogo “Monopólio” é um dos mais conhecidos e jogados da actualidade. Alia elementos aleatórios, introduzidos por dois dados, a decisões fundamentadas baseadas em estratégias de negócio. A história do “Monopólio” remonta a 1904, quando Elisabeth Magie criou o jogo “The Landlord’s Game”, com o intuito de ensinar as complexidades do monopólio de terras e propriedades (Albertarelli 2000).

O jogo foi entretanto desenvolvido, tendo vindo a ser comercializado pela Hasbro, e distribuído posteriormente pelo mundo inteiro, sendo que há versões traduzidas para várias línguas, incluindo o português. A versão original portuguesa com a divisa antiga deu recentemente lugar a uma nova versão, em que a moeda de troca, seguindo o desenvolvimento da realidade, deu lugar ao Euro.

Correntemente o desenvolvimento de jogos de tabuleiro é um grande negócio, e a criação destes jogos é para alguns uma profissão a tempo inteiro. São organizados congressos e palestras sobre o tema, e há um grande número de adeptos que se juntam regularmente para sessões de jogo, criando mesmo em certos casos torneios a nível nacional. Além de jogos baseados em ideias originais, muitos clássicos do cinema ou da literatura dão origem a jogos de tabuleiro, à semelhança daquilo que se passa com os jogos de computador ou consola. O jogo “Senhor dos Anéis” é baseado na trilogia de J. R. R. Tolkien e é um dos jogos mais conhecidos da actualidade.

O jogo “Dungeons and Dragons” representa exactamente a situação inversa. Publicado pela primeira vez em 1974 pela Tactical Studies Rules, Inc. (TSR) e criado por Gary Gygax e Dave Arneson, este jogo alcançou um sucesso de tal ordem que deu origem a uma grande produção cinematográfica, uma série animada de televisão, figuras coleccionáveis e a um jogo de computador. Existe também um MMORPG (*massively multiplayer on-line role playing game*), um jogo online para múltiplos utilizadores.

Vários jogos de tabuleiro foram recriados para computadores, consolas e telemóveis, tal é o seu impacto na comunidade de jogadores. Monopólio, Risk, Dungeons and Dragons e Snakes and Ladders são exemplos de jogos que foram transportados para dispositivos electrónicos. Não só as versões quase 100% fiéis ao jogo original foram criadas, mas também jogos novos foram criados tendo como base a ideia original de um jogo de tabuleiro. Foram também desenvolvidos jogos fiéis às regras dos jogos de tabuleiros, incluindo no factor aleatório introduzido pelo lançamento de dados e no movimento por turnos casa a casa, que incluem elementos não possíveis no tabuleiro. Exemplos deste tipo de jogos são o “M&M's Minis' Blast”, ou o excelente jogo “get the glass”, desenvolvido como parte de uma campanha publicitária da California Milk Processor Board (CaliforniaMilkProcessorBoard 2007).



Fig. 3 – M&M's Mini's Blast

Em Portugal, a comunidade de RPG (*Role Playing Games*) e Jogos de Tabuleiro divulga notícias no seu portal (<http://www.abreojogo.com/>), organiza sessões e entrevistas.

Prevê-se durante o próximo ano o lançamento do primeiro jogo português de projecção internacional. “Vinícola” é um jogo que explora um dos grandes mercados de exportação português, o vinho. Criado por Vital Lacerda, o jogo será publicado pela italiana “What’s your game”, e consiste na produção de vinhos de qualidade que poderão ser trocados por dinheiro ou por “pontos de vitória”. Os melhores vinhos são enviados para a feira dos vinhos onde podem atingir a fama e ganhar prémios (Boardgamegeek 2009).



Fig. 4 – Vinícola, de Vital Lacerda – O primeiro jogo português de projecção internacional.

2.3 Jogos electrónicos

Nos inícios da década de 60, o jogo “Periscope-Firing Range” foi aquele que é tido como o primeiro jogo portátil electromecânico alimentado por baterias. Apesar de não gozar da mobilidade

das consolas que o seguiram, pelo seu volume e peso, o jogo podia ser transportado e jogado em qualquer lugar. Vinha acompanhado de um tripé no qual podia ser montado e as suas regras eram extremamente simples. Imagens passavam no fundo e o jogador devia carregar no botão de disparo quando um navio se encontrasse na linha de fogo. Os sons que produzia eram de origem mecânica e não eram representativos de um tiro certo. Uma luz aparecia também por trás dos navios quando um tiro era disparado (Handheldmuseum 2005).



Fig. 5 – Cragstan Periscope-Firing Range

O ano de 1972 foi marcado pelo lançamento de um dos primeiros antecessores das consolas portáteis que conhecemos hoje. O “Electro Tic-Tac-Toe” da Waco Electro era um jogo electromecânico, constituído por luzes que acendiam e apagavam conforme as acções dos jogadores (DeMaria and Wilson 2003). A tecnologia da altura não permitia a utilização de soluções de inteligência artificial necessárias para que um jogador pudesse jogar o jogo sozinho, pelo que um adversário era condição essencial. Apesar de ser extremamente rudimentar pelos padrões de hoje, esta máquina foi a rampa de lançamento para os jogos que se seguiram (DeMaria and Wilson 2003).



Fig. 6 – Waco's Tic-Tac-Toe

As primeiras grandes evoluções neste campo surgiram entre 1976 e 1977 com o lançamento de uma série de jogos pela conhecida “Mattel”. O primeiro jogo completamente electrónico foi o “Auto Race”, seguido pouco tempo depois pelo “Football” (DeMaria and Wilson 2003). Devido ao facto de utilizarem LEDs (*Light Emitting Diodes*) em vez de lâmpadas, estes jogos podiam ser alimentados apenas com uma pilha de 9 volts, ainda que os primeiros modelos viessem também acompanhados de um adaptador que permitia a ligação à rede eléctrica.



Fig. 7 – Mattel Auto Race e caixa

Apesar do seu tempo de vida extremamente reduzido, de apenas 2 anos, a consola Microvision trouxe em 1979 alguns dos elementos necessários para o desenvolvimento deste tipo de aparelhos. A introdução dos ecrãs de LCD permitia reduzir o tamanho das baterias e aumentar a autonomia, enquanto a utilização de *cartridges* com diferentes jogos (DeMaria and Wilson 2003) possibilitava a utilização do mesmo aparelho para a execução de vários jogos, diminuindo o custo e aumentando a margem de lucro. Infelizmente, devido ao pequeno tamanho do ecrã, aos problemas que surgiam com a consola e à escassez de títulos jogáveis, esta consola retirou-se em 1981.



Fig. 8 – Consola Microvision

A partir de 1980, as mais conhecidas consolas começaram a surgir, tendo atingido uma fama ainda hoje reconhecida. A Nintendo lançou uma série de jogos chamados Game & Watch (DeMaria and Wilson 2003), inicialmente com mecânicas de jogo extremamente básicas mas que foram evoluindo com o tempo. Também o formato destas consolas foi sendo alterado, e vários tipos de desenho surgiram no mercado, desde os mais simples aos muito imitados “clamshell”, patentes em muitos telemóveis de hoje e em consolas como a Gameboy Advance SP ou a futura OpenPandora.

Com a utilização da tecnologia LCD que estava em voga nas calculadoras da altura, estas consolas apresentavam também um relógio digital e, devido ao seu baixo consumo, podiam manter-se sempre ligadas sendo alimentadas por uma ou duas pilhas de relógio, como a conhecida LR44.

Vários conceitos e personagens desta época foram utilizados em gerações futuras da Nintendo. O botão em forma de cruz patenteado pela marca era o mesmo que mais tarde viria a ser utilizado na consola de sala Nintendo Entertainment System, ou NES. Personagens como Mário ou Donkey Kong eram já os astros de jogos das consolas Game & Watch, continuando presentes em todas as gerações futuras das consolas desta empresa.

2.4 Videojogos

Desde as peças num tabuleiro dos jogos clássicos às luzes e LEDs dos jogos electrónicos, sempre foi fundamental transmitir o resultado das acções do jogador e os jogos sempre necessitaram de elementos de interactividade. Na década de 70 os receptores de televisão começaram a ser usados de formas alternativas, com a introdução de gravadores de vídeo, que coexistiram temporariamente com os *LaserDisks* e acabaram por evoluir para DVDs. O ecrã passou a transmitir informações que não as enviadas pelas emissoras de TV, e a utilização deste para *feedback* de um jogo era uma possibilidade com bastante potencial.

O Magnavox Odyssey foi o primeiro sistema de jogos de vídeo a entrar nos lares. Era um sistema analógico alimentado por baterias, não produzia qualquer som e os jogos podiam ser jogados com os controladores incluídos ou com uma pistola "*Light Gun*". Baseado no protótipo criado em 1968 por Ralph Baer, o sistema foi finalmente lançado em 1972 depois de ter sido licenciado à companhia Magnavox. Esta empresa deixou passar a ideia de que o sistema só funcionava nos seus receptores de televisão, o que fez baixar as vendas do sistema em vez de subir as vendas de televisores Magnavox, como era esperado (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 9 – Magnavox Odyssey

Apesar de não ser o primeiro sistema de jogos na televisão, o Atari Pong foi talvez o mais famoso e imitado. Com base na máquina *Arcade* do mesmo nome, o engenheiro Harold Lee, da Atari, começou em 1973 a modificar e miniaturizar o *hardware* para que este pudesse ser usado em casa. Em 1975 este sistema começou a ser comercializado pela Sears (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 10 – Atari Pong

Depois do relativo sucesso do Odyssey, a Magnavox lançou dois sistemas mais actuais, em 1975. O jogador marcava a pontuação com apontadores de plástico montados nos sistemas Odyssey 100 e Odyssey 200, embora o modelo de gama mais alta tivesse já um arcaico sistema de contagem da pontuação. Este último modelo acrescentava um terceiro jogo, de seu nome “SMASH” aos dois jogos disponibilizados pelo modelo mais barato (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 11 – Odyssey 100 e Odyssey 200

O ano de 1976 viu o lançamento do Atari Super Pong. No seguimento do enorme sucesso alcançado pelo Pong original, esta nova versão permitia ao jogador seleccionar uma de quatro versões diferentes do jogo. Embora muitas versões do mesmo jogo fossem surgindo por parte de outros fabricantes, a Atari manteve-se no topo devido ao renome e sucesso alcançado com as suas máquinas de *Arcade*. Os sistemas da Atari eram também conhecidos pelo uso da cor e sons de qualidade superior (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 12 – Atari Super Pong

A empresa Coleco começou por ser uma empresa do ramo de couro para sapatos. O seu nome provém de junção das palavras “Conneticut Leather Company”, mas quando o seu principal ramo de negócios passou à moldagem de plástico e a parte do tratamento de couro foi vendida, a empresa entrou noutro ramo que faria o seu nome entrar para a história. As consolas Coleco Telstar e a mais conhecida ColecoVision foram marcos de sucesso. Vários modelos da Telstar foram produzidos entre 1976 e 1978 com diferentes jogos, introdução da cor no modelo Telstar Colormatic, diferentes métodos de controlo incluindo a *Light Gun* e um sistema de *cartridges* na Telstar Arcade. A Telstar original era vendida nos Estados Unidos por \$50, o que praticamente por si só justificava a fama alcançada pela consola (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 13 – Coleco Telstar

Numa altura em que o jogo Pong era tão famoso que a Texas Instruments criou um circuito intitulado “Pong on a chip”, a Magnavox voltou em força ao mercado com os sistemas Odyssey 300, 400 e 500. Com pequenas variações sobre os Odysseys originais, o modelo 400 mostrava um “W” de *Winner* do lado do vencedor, e mostrava as pontuações quando a bola desaparecia. Em 1977 foram

lançados os sistemas 2000, 3000 e 4000, tendo o sistema 4000 sido o último aparelho com o Pong lançado pela Magnavox (DeMaria and Wilson 2003; Thegameconsole.com 2006).

O Fairchild Channel F nunca chegou a ter o sucesso que merecia, mas foi o primeiro sistema a utilizar *cartridges* contendo uma memória ROM com código em vez de circuitos dedicados, o que mais tarde viria a ser copiado e tornado normal por outros fabricantes. Um total de 26 *cartridges* foram lançadas, contendo até 4 jogos cada. As *cartridges* eram numeradas, num golpe de *marketing* que visava levar o consumidor a completar uma colecção (DeMaria and Wilson 2003; Thegameconsole.com 2006).

A Atari voltou mais uma vez, em 1977, a ditar as novas normas, com uma consola que viria a tornar-se extremamente popular. Ainda nos dias de hoje uma legião de *retro-gamers* utiliza emuladores do sistema VCS 2600, que viria em 1982 a responder pelo nome de Atari 2600. Com este equipamento, o sistema de jogos em *cartridges* começou realmente a tornar-se popular e tem vindo a ser usado até aos dias de hoje. O Atari 2700 nunca chegou a ser lançado, e o Atari 2800 lançado no Japão em 1983 não alcançou a mesma popularidade devido à forte concorrência do Nintendo Famicom, ou Nintendo Entertainment System, como ficou conhecido na Europa (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 14 – Atari 2600

Em Portugal, o sistema mais famoso da década de 70 foi o Philips Videopac, lançado em 1978. Esta consola era a versão Europeia da Magnavox Odyssey², também chamada em outros países de Philips Odyssey ou Philips Odyssey². Os cartões que utilizava alternavam entre diferentes jogos, e a consola estava munida de um teclado em forma de membrana que permitia a utilização de jogos educacionais ou programação básica. Uma inovação muito apreciada na altura foi também o módulo que permitia a síntese de voz e efeitos de som e música melhorados (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 15 – Odyssey² / Philips Videopac G7000

Várias consolas foram lançadas pela Coleco, Atari e Magnavox, introduzindo novos jogos e elementos como o uso da côm, mas o grande marco no fim da década foi a consola da Mattel Electronics. Lançada em 1980 com o nome Mattel Intellivision, a empresa que a vendia pretendia destronar o Atari 2600. A consola Intellivision utilizava melhores componentes do que a concorrência, com melhores gráficos e melhor som, mas o que a tornou um sucesso foi a promessa de um teclado que permitiria transformar a consola num computador pessoal de baixo custo. O teclado acabou por não ser lançado por ter um custo elevado, e quando finalmente foi comercializado por pressão do governo já não teve a aceitação que poderia ter tido. A Intellivision II viria a ser lançada em 1982 (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 16 – Mattel Intellivision

O sistema Vectrex, lançado em 1982, trouxe duas novidades: O monitor incluído (não era necessário ligar à televisão) e os gráficos vectoriais. Enquanto a primeira não teve grandes herdeiros, a segunda foi o princípio dos jogos em três dimensões a que estamos habituados nos dias que correm. Como o monitor não variava de dimensões, os jogos do Vectrex permitiam simular a cor com uma camada colorida estática que se sobrepunha aos gráficos gerados pela consola. Uma grande parte dos jogos das *Arcade* do princípio da década de 80 utilizava gráficos vectoriais, e a Vectrex deu ao consumidor a possibilidade de jogar esses jogos em casa. Tal como o Atari 2600, o Vectrex continua a ser um sistema bastante emulado, mais de vinte e cinco anos após o seu lançamento (DeMaria and Wilson 2003; Thegameconsole.com 2006).

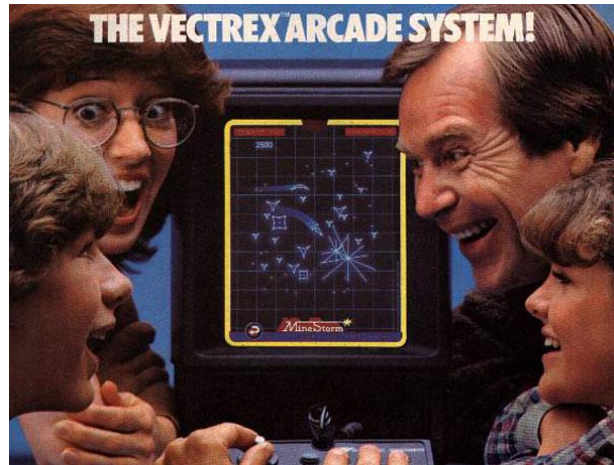


Fig. 17 – Vectrex

O ano de 1982 viu também o lançamento da terceira geração de consolas da Coleco. A ColecoVision trouxe consigo várias novidades que a tornaram famosa: gráficos ao nível das máquinas disponíveis nos salões de jogos, botões programáveis nos comandos, possibilidade de expansão e a capacidade de jogar jogos de outros sistemas. Parte do sucesso da ColecoVision foi também o facto de esta ter sido a primeira consola a disponibilizar o famoso jogo “Donkey Kong”, da Nintendo. Um módulo para esta consola permitia jogar jogos do Atari 2600, o que a tornava ainda mais apetecível. Após uma disputa em tribunal com a própria Atari, e tendo a Coleco saído vencedora, a empresa aproveitou para lançar também a Coleco Gemini, uma consola compatível com a Atari 2600 (DeMaria and Wilson 2003; Thegameconsole.com 2006).



Fig. 18 – ColecoVision

A época dos jogos de TV foi dando origem à das verdadeiras consolas em meados da década de 80, com o lançamento de uma das mais carismáticas consolas de todos os tempos, a Nintendo Famicom (Nintendo Family Computer), conhecida em Portugal e na Europa como Nintendo Entertainment System (DeMaria and Wilson 2003; Thegameconsole.com 2006).

2.5 Consolas de jogos

Consolas de jogos não são muito mais do que os videojogos apresentados na secção anterior aos quais foi atribuído um novo nome. A distinção entre videojogos e consolas de jogos é relativamente livre, sendo que as consolas são equipamentos criados com o intuito de jogar jogos de video. Em certo momento da história o mercado estava criado e os videojogos deram origem às consolas que continuam a vender em enormes quantidades nos tempos correntes.

Começando pelas consolas de sala, ligadas à televisão, os avanços na tecnologia, principalmente no que diz respeito à miniaturização dos componentes, criação de circuitos específicos para a realização de funções complexas, ecrãs dedicados com funcionamento a pilhas e menores consumos de energia viriam a dar origem a consolas portáteis.

2.5.1 Consolas de Sala

A Nintendo, que começou por ser uma produtora de jogos, é agora conhecida internacionalmente pelas suas consolas. Após o sucesso de alguns jogos desta produtora no início da década de 80, e dada a expansão dos sistemas de jogos de vídeo que se sentia na altura, a Nintendo optou por entrar no negócio das consolas. Em 1983 foi lançada no Japão a Nintendo Famicom. Após alguns problemas iniciais, o desenvolvimento de uma nova *board* fez com que no final de 1984 esta consola alcançasse o estatuto de consola mais vendida no Japão, no ano de 1984. No ano seguinte, e com base no sucesso alcançado, esta consola entrou no mercado Americano com uma nova cara e um novo nome, Nintendo Entertainment System (NES). O sucesso alcançado no Japão viu-se repetido internacionalmente. A NES tornou-se um dos mais importantes marcos na história deste tipo de equipamentos e serviu como referência para as gerações seguintes. Apesar de que alguns jogos de vídeo anteriores (como o Atari 2600) poderem ser consideradas como consolas, o acrónimo NES é ouvido como a primeira verdadeira consola da história (Thegameconsole.com 2006).



Fig. 19 – Nintendo Entertainment System

Como a Nintendo, a Sega criou no início da década de 80 alguns jogos populares e também viu na produção de consolas uma grande oportunidade. Com base no sucesso da Entertainment System, a Sega decidiu melhorar uma consola que lançara em 1983 no Japão, a SG-1000, e avançar com ela para o mercado norte-americano. A Sega Master System saiu em 1986 e como fora previsto, foi um sucesso de vendas. Tecnicamente superior à sua concorrente mais directa, a Master System oferecia também um leque de acessórios inovadores e a possibilidade de utilizar cartões com jogos, semelhantes às *cartridges* mas mais baratos e com menos código. A falta de jogos interessantes para a utilização destes acessórios limitou o seu sucesso. Enquanto a Master System não chegou ao sucesso da NES nos Estados Unidos, na Europa foi uma consola de grande êxito. Em 1990 a Master System II, mais pequena e mais barata, sucedeu à original sem grande sucesso (Thegameconsole.com 2006).



Fig. 20 – Sega Master System

Três anos passados sobre o lançamento da Master System, a Sega viu a oportunidade de se antecipar a outros fabricantes no lançamento de uma consola de nova geração. Apesar do gigante NEC ter lançado a Turbo Grafx 16, com um processador gráfico de 16 bits capaz de gerar 256 cores e som de seis canais, o mercado estava receptivo e na Europa a consola da NEC não tinha tido grande aceitação. A Mega Drive, ou Sega Genesis como ficou conhecida em alguns mercados corrigiu o erro anterior da Sega e trouxe excelentes jogos da marca e de outras empresas, assegurando o seu sucesso. A Mega Drive foi, nos anos que se seguiram, a base para alguns acessórios que disponibilizavam jogos de 32 bits utilizando a Sega 32X ou com um máximo de 650MB num suporte novo e barato, o CD, por intermédio da Sega CD, ambos com muito pouca saída por falta de grandes melhoras e preço elevado (Thegameconsole.com 2006).



Fig. 21 – Sega Mega Drive

A SNK Neo-Geo surgiu em 1989 e a consola doméstica AES era uma réplica em termos de funcionalidades das máquinas de *Arcade* MVS que utilizavam o mesmo *hardware*. Munida de um processador Z80 e um Motorola 68000 e com 330MB de espaço de armazenamento, apresentava gráficos coloridos e fluidos bastante acima da concorrência. Os preços tanto da consola como dos jogos eram também bastante superiores aos dos seus rivais, o que limitou o seu sucesso internacionalmente e a fez passar praticamente despercebida na Europa. Em 1994 o lançamento do Neo-Geo CD conseguiu tornar os jogos mais baratos devido ao preço do suporte, mas os tempos de carga aumentaram bastante (Thegameconsole.com 2006).



Fig. 22 – SNK Neo Geo com controlador

Em 1991 a Nintendo voltou à carga e a entrar na história com a Super Nintendo Entertainment System. Apesar de não chegar onde chegou a NES na América e na Ásia, devido à força que a consola da Sega tinha na altura, a SNES foi um vendedor de topo em todo o mundo. Foi a consola de 16 bits mais vendida no planeta, principalmente depois de a Sega se retirar deste segmento. Como com outras consolas da Nintendo, uma grande parte do seu sucesso deveu-se aos jogos inteligentes e imaginativos mais do que ao *hardware*. Ainda nos tempos actuais os jogos desta

consola servem como referência e continuam a ser jogados em emuladores ou mesmo em consolas originais (Thegameconsole.com 2006).



Fig. 23 – Super Nintendo Entertainment System ou SNES

No fim da década de 80 os jogos já eram distribuídos em versões para consolas e computadores pessoais. O sucesso de computadores como o Commodore Amiga e o Atari ST afastou muitos potenciais compradores das consolas, mas o mercado estava em desenvolvimento. Com a experiência ganha com o ST aliada às suas consolas anteriores, a Atari lançou em 93 e 94 aquela que comercializou como a primeira consola de 64 bits, Atari Jaguar. As vendas andaram longe das conseguidas pela Sega ou Nintendo, mas a Jaguar tornou-se uma consola de culto entre os utilizadores de jogos *homebrew*. A escassez de jogos comerciais fez no entanto com que a consola se retirasse cedo do mercado (Thegameconsole.com 2006).



Fig. 24 – Atari Jaguar

Em 1995 havia versões de consolas ou acessórios de leitura de CDs para praticamente todos os equipamentos que tinham tido algum sucesso, e a Nintendo encomendou à Sony uma unidade de CDs para a SNES. A Sony começou a trabalhar no projecto, mas o alto preço cobrado por esta empresa fez com que a Nintendo optasse por trocar de parceiro e trabalhar com a Philips. A Sony aproveitou o trabalho que já tinha desenvolvido para criar a sua própria consola, e em 1994 a Playstation foi lançada nos Estados Unidos, e no ano seguinte na Europa. A Playstation utilizava, além do processador principal, uma GPU (*Graphics Processing Unit*) que lhe permitia cálculos de polígonos 2D e transformações 3D e mapeamento de texturas, trazendo um alargado leque de jogos 3D (Thegameconsole.com 2006).



Fig. 25 – Sony PlayStation

Após a tentativa de entrada no mercado dos jogos 3D pela Sega, com a Saturn (consola que teve muito pouca saída), a Nintendo atacou com a Nintendo 64 que, apesar de não chegar à fama das unidades anteriores da marca, saiu-se bastante melhor do que a rival Sega. Lançada em 1996 e 1997 em diferentes partes do globo, esta consola divulgou o jogo Super Mario 64, a primeira aparição em 3D do famoso canalizador (Thegameconsole.com 2006).



Fig. 26 – Nintendo 64

Sem dúvida a consola mais carismática e adorada pelos programadores independentes e a comunidade *homebrew* foi a Sega Dreamcast. Tecnologicamente avançada, fácil de programar e com excelentes jogos, a Dreamcast fechou com chave de ouro a era das consolas da Sega. Apesar das excelentes vendas no primeiro ano de produção, entre 1998 e 1999, o impacto que teve o anúncio e o lançamento da Playstation 2 acabou por levar a marca a abandonar o mercado das consolas, dedicando-se exclusivamente à criação de jogos (Thegameconsole.com 2006).



Fig. 27 – Sega Dreamcast

Depois do sucesso da Playstation, o público estava curioso e ansioso por ver o que a Sony traria em seguida. No ano 2000 a Sony apresentou ao mundo a Playstation 2 (Thegameconsole.com 2006), ou PS2 como é mais conhecida. Ainda hoje encontrada em muitos lares na sua versão original ou *Slim*, a PS2 foi a consola que mais rapidamente atingiu os 100 milhões de unidades, tendo alcançado este feito em quase metade do tempo da sua antecessora (Sony 2005). Mesmo depois do lançamento da sua sucessora, a PS2 continuou a vender, sendo ainda hoje procurada no nosso país por causa de títulos como “Buzz: Mega Quiz”.



Fig. 28 – Sony PlayStation 2

O realismo começava a complementar a jogabilidade, e a Gamecube da Nintendo enfrentou sérios problemas nesta área quando comparada com as rivais da Sony ou da Microsoft. Apesar de manter a compatibilidade com títulos da consola anterior, a Gamecube não tinha também o leitor standard de DVD das consolas da concorrência da altura, o que não permitia ver filmes ou ouvir música. Os seus jogos eram distribuídos em DVDs de formato proprietário com o objectivo de combater a pirataria, mas com a quebra da protecção esta medida acabou por ter mais resultados negativos do que positivos. Apesar de a Sega Dreamcast ter sido a primeira consola a oferecer *hardware* de ligação à rede incluído (na altura um modem analógico) a experiência *on-line* não teve grande aceitação. O jogo *on-line* começava a ser muito procurado nos anos de 2001 e 2002, altura em que a Gamecube foi lançada, e a Nintendo não oferecia as soluções dos seus grandes rivais. Como tal, mais uma geração de consolas desta marca ficava para trás em relação à concorrência, e a Nintendo teve que aguardar uns anos antes de voltar ao topo como fabricante de consolas de sala (Thegameconsole.com 2006).



Fig. 29 – Nintendo GameCube

A gigante Microsoft entrou com o pé direito nesta área de mercado. Após ter colaborado com a Sega que fornecia o Sistema Operativo da Microsoft Windows CE na sua última consola, a empresa criou a Xbox baseada na arquitectura dos computadores aos quais fornecia o seu famoso Sistema Operativo. Além do aspecto impressionante, a imagem criada com estratégias de marca e estreias no mundo das consolas como o disco rígido em vez de cartões de memória, a Microsoft ganhou à concorrência uma grande fatia de mercado. Jogos exclusivos como o excelente “Halo: Combat Evolved” ajudaram também esta consola a cativar utilizadores. Um novo mercado foi aberto também para a divisão de desenvolvimento de jogos da Microsoft, o “Microsoft Game Studio” (Thegameconsole.com 2006).



Fig. 30 – Microsoft Xbox

Presentemente o mercado é dividido entre os três concorrentes de topo, que se aguentaram firmemente após o início e fim de muitas consolas. A Microsoft antecipou-se com a sua Xbox 360, que apesar de ser tecnicamente inferior à oferta corrente da Sony – a Playstation 3 – conta com jogos visualmente semelhantes e de jogabilidade igual senão melhor do que a da principal concorrente. Ambas as consolas executam jogos quase fotorrealistas em alta definição e com som *surround*. A Nintendo, consciente da forte concorrência, optou pelo lançamento de uma consola que tecnicamente se posiciona uma geração atrás das duas rivais, mas num golpe característico conquistou publico com a sua jogabilidade complementada pelo inovador sistema de

controlo da Wii, o Wiimote. Pela primeira vez foram abandonados (ou complementados) os botões e *joysticks* por um comando que responde a movimentos naturais (Thegameconsole.com 2006).



Fig. 31 – Wiimote e Nintendo Wii, Sony Playstation 3 e Xbox 360

Numa época em que o jogo *on-line* é um dos objectivos do jogador, em que os gráficos se aproximam do fotorrealismo e em que os métodos de controlo estão a ser substituídos por outros mais naturais (o Wiimote, o anunciado Project Natal da Microsoft ou o método patenteado pela Sony de utilização de objectos do dia a dia para controlo de jogos), assiste-se a um regresso às origens que completa estas experiências. A Sony apresenta jogos de plataformas como “Little Big Planet”, a Nintendo mantém sempre actual o seu herói Mário no “New Super Mario Bros”, e a Microsoft disponibiliza o “Xbox Live Arcade”, uma área do Microsoft Live onde é possível adquirir jogos como “Braid”, um *platformer* inovador.

Apesar da procura de novas experiências imersivas, realidade virtual e aumentada e fotorrealismo, o jogo de *arcade* não está morto e continua tão forte como sempre. Nunca houve tanto dinheiro investido neste mercado que continua em crescimento, e a inovação pode ser, e é, aliada aos conceitos clássicos dos videojogos para criar experiências extremamente recompensadoras (Thegameconsole.com 2006).

2.5.2 Consolas Portáteis

Em 1989, uma década passada sobre o lançamento da Microvision, a Nintendo lançou o Game Boy, uma evolução colossal sobre aquela consola. Com um ecrã LCD maior e de mais alta resolução, autonomia alargada e uma enorme colecção de títulos disponíveis, o Game Boy obteve um sucesso a nível mundial de que poucas consolas se podem orgulhar. Esta consola foi também uma das primeiras a dispor de diferentes acessórios, como o cabo de dados que permitia jogar jogos em conjunto com outros jogadores, ou uma câmara cujas fotografias eram impressas numa impressora térmica desenvolvida também para a consola. Agora com vinte anos de existência, o Game Boy original é ainda encontrado em vários lares e, apesar de tecnicamente ultrapassado,

continua a ser jogado por muitos devido à enorme panóplia de títulos disponíveis e à excelente jogabilidade pela qual a Nintendo é conhecida desde sempre (DeMaria and Wilson 2003).



Fig. 32 – Nintendo Game Boy

Algumas das características apontadas à partida como negativas no Game Boy foram exactamente as que levaram ao fracasso o seu concorrente do mesmo ano, o Atari Lynx. Esta consola era em muitos aspectos bastante superior à da Nintendo. Ecrã a cores, retro iluminado e maior poder de processamento que incluía escalamento de imagens por *hardware* fizeram com que a consola fosse grande, cara e com pouca autonomia. As agressivas campanhas de *marketing* da Nintendo ajudaram a fazer com que esta consola quase caísse no esquecimento, apesar de ter sido um dos marcos importantes na história das consolas portáteis, pela sua inovação (DeMaria and Wilson 2003).



Fig. 33 – Atari Lynx

No fim da década de 80, a Sega era a principal concorrente da Nintendo na criação de jogos e consolas. O contra-ataque à Game Boy deu pelo nome de Game Gear, e como quase todos os produtos da Sega tinha todos os argumentos para vencer.

A Game Gear era praticamente uma Master System (uma consola de sala da Sega) portátil com um ecrã de mais baixa resolução mas com um número de cores alargado. Era fácil recriar jogos para esta consola por serem praticamente os mesmos da Master System. A consola apresentava um ecrã a cores como a Lynx, mas tinha um preço mais baixo e uma muito maior gama de *software*, cerca de trezentos e noventa títulos. A sua autonomia, no entanto, não se comparava à do Game Boy. Num formato bastante maior, alojava seis pilhas que permitiam um tempo de jogo de cerca de metade daquele que o Game Boy apresentava com apenas quatro. O seu preço era também quase o dobro do da consola da Nintendo, e nesta altura o público não estava habituado a gastar tanto dinheiro neste tipo de aparelhos como nos tempos actuais.

Vários acessórios foram lançados para a Game Gear, de entre os quais o mais conhecido e que mais furor fez na época terá sido o sintonizador de TV, que transformava a consola num receptor de televisão portátil (DeMaria and Wilson 2003).



Fig. 34 – Sega Game Gear

Em 1998 a Nintendo lançou a Game Boy Color ou GBC. A cor foi a grande novidade nesta consola que duplicou também o poder de processamento e a quantidade de memória do original. Esta nova versão era retro compatível e permitia jogar os novos jogos a cores mantendo também a possibilidade de correr a larga biblioteca de títulos que existiam já no mercado (DeMaria and Wilson 2003). Em combinação com o modelo original, a Nintendo vendeu 118.69 milhões de unidades em todo o mundo (Nintendo 2005).

Outras consolas surgiram, como a NeoGeo Pocket (e posteriormente a NeoGeo Pocket Color) ou a Wonderswan (com a sua versão a cores a chegar também ao consumidor no ano 2000). Estas consolas não tiveram, no entanto, grande aceitação na Europa.

Já na década corrente, a Nintendo voltou a apresentar um produto que se viria a tornar um dos mais utilizados no nosso mercado. O Game Boy Advance, ao contrário do Game Boy Color,

mostrou uma grande evolução em relação ao original. Com um processador adaptado às novas tendências, dois botões posteriores e um ecrã maior, os jogos para GBA estão entre os mais jogados pelos jogadores que utilizam actualmente as consolas *Open Source* nas quais se centra este documento, por meio do emulador GpSP - de Exophase - para a Sony PSP ou a GP2X, ou ainda para o iPod Touch ou iPhone.

A Game Boy Advance SP e mais tarde a Game Boy Micro foram novas encarnações desta consola, a primeira em formato “*Clamshell*” e a segunda num formato mais pequeno, que podia facilmente ser transportado em qualquer bolso. Mais de 80 milhões de unidades dos três modelos combinados foram vendidas em todo o mundo (Nintendo 2008).



Fig. 35 – Game Boy Advance Original, SP e Micro

A Nokia fez a sua entrada no mundo das consolas com a N-Gage. Adorada por muitos e odiada por outros tantos, a grande inovação desta consola foi a junção de vários equipamentos num só aparelho, tendência que tem vindo a crescer desde então. Infelizmente, como telefone era muito pouco prático – o interlocutor posicionava o aparelho perpendicularmente à face – e como consola ficava bastante aquém das ofertas da Nintendo. A Nokia voltou à carga com uma revisão – a N-Gage QD – que ainda assim não trouxe grandes revoluções. Correntemente o N-Gage deixou de ser uma linha de consolas físicas para se transformar numa plataforma de jogos que funciona em vários terminais da marca, não tendo no entanto a ergonomia necessária a uma consola de jogos.



Fig. 36 – Nokia N-Gage e N-Gage QD

Com a adaptação de Sistemas Operativos de PDAs para várias funções, duas consolas surgiram. A TapWave Zodiac corria uma versão de Palm OS 5 sendo simultaneamente um PDA e uma consola de jogos. A Gizmondo, por seu lado, continha praticamente tudo o que era possível introduzir num aparelho daquelas dimensões: GPS, comunicações GSM, música e vídeo, e os controlos necessários para jogar. Devido a más decisões do CEO no último caso, e à queda do Sistema Operativo Palm OS, ambas as consolas acabaram por cair praticamente no esquecimento quando as empresas que as produziam declararam falência (AllExperts.com).

À excepção das consolas *Open Source* nas quais este documento se centra, visitadas no capítulo 3, o mercado actual de consolas portáteis é partilhado entre as duas empresas líderes de vendas: A Nintendo e a Sony. Ambas foram disponibilizadas ao público no ano de 2004, e ainda hoje coexistem de forma mais ou menos pacífica, pelos ligeiramente diferentes mercados que tentam atingir.

A Nintendo DS, ou NDS, apresenta várias inovações como é já habitual nos produtos da marca. Em vez de tentar vender pelos impressionantes gráficos 3D, a Nintendo ofereceu uma consola com dois ecrãs, um deles sensível ao toque, e utilizou este último e mesmo o microfone como controladores adicionais para os jogos. Aquilo que poderia de outra forma ser apenas uma inovação técnica ou prova de conceito, foi pelas mãos da Nintendo transformado em várias formas extremamente divertidas e aliciantes de interagir com os jogos. Reconhecimento de voz e detecção de vento (o utilizador sopra para a consola para controlar certas situações) trazem uma jogabilidade excelente, como só a Nintendo sabe fazer.

Além do maior poder de processamento e das características já referidas, a NDS contém também um módulo de comunicações sem fios que permite o acesso à *internet*, existindo mesmo um navegador da casa Opera com o qual é possível interagir de forma engenhosa. É também viável, por meio destas comunicações sem fios, manter conversas com outro utilizador da NDS, com texto ou mesmo gráficos e desenhos, ou interagir com a última consola de sala da Nintendo, a Wii.

A NDS permite jogar jogos do Game Boy Advance, mantendo a retro compatibilidade conhecida da Nintendo e alargando o número de jogos disponível. A NDS deu lugar à NDS Lite, uma versão mais pequena mas sem outras alterações de maior. Recentemente foi lançada a Nintendo DSi, com ecrãs maiores e duas câmaras VGA. O conector que permitia a utilização de *cartridges* de GBA foi removido, dando origem a uma entrada de cartões SD. Enquanto esta medida acompanha as tendências do mercado, substituindo progressivamente os conteúdos em suporte físico por conteúdos digitais descarregáveis, a retro compatibilidade foi infelizmente afectada. Ainda que a Nintendo disponibilize os jogos de GBA na internet, cartões com funções especiais, como o Rumble Pack, não poderão ser usados.



Fig. 37 – Nintendo Dsi Branca e Preta

A Sony entrou directamente para uma posição de topo no mercado das consolas portáteis, por meio da sua Playstation Portable. Apesar de alguns problemas como pixéis mortos no ecrã ou o rasto deixado pelos movimentos rápidos, o poder de processamento 3D acima da concorrência fez da PSP uma consola que ainda hoje se pode considerar actual. Os jogos tridimensionais não foram a única característica que a PSP herdou da Playstation original. O controlador analógico, ainda que longe da perfeição, trouxe às consolas portáteis a jogabilidade que até então só era possível nas consolas de sala como a Playstation, Dreamcast ou Nintendo 64. O ecrã de 4.3 polegadas foi o maior disponibilizado numa consola portátil, e acompanhado do GPU (*Graphics Processing Unit*) a 166MHz e 2MB de memória *on-board* permitiu que os jogos 3D com impressionantes efeitos especiais acompanhassem os adeptos desta consola onde quer que eles fossem (Nix 2005).

Tal como a Nintendo DS, a PSP utiliza também uma ligação WiFi com a qual o utilizador pode navegar na internet. A PSP pode também partilhar e complementar conteúdos da consola de sala da Sony, a Playstation 3.

A PSP original deu origem à versão Slim & Lite, mais pequena e leve, mas sem grandes alterações à excepção da saída para televisão, carregamento da bateria por USB e uma duplicação da memória RAM, de 32MB para 64MB. Posteriormente, a versão PSP-3000 trouxe ainda um ecrã melhorado e um microfone incorporado. A grande evolução desta consola está no entanto prevista para Outubro do ano corrente. A PSP Go será ainda mais pequena, com um ecrã de 3.8 polegadas em vez das 4.3" do original, um peso 43% mais baixo e funcionalidades *Bluetooth*. A maior novidade será o abandono dos discos UMD (Universal Media Disk) em prol de uma capacidade de armazenamento interno de 16GB e cartões Memory Stick Micro (que virão substituir os Memory Stick Duo utilizados na original). Os jogos serão descarregados da Playstation Store, num procedimento que começa a transformar-se em padrão (Parrish 2009).



Fig. 38 – Sony PSP Sim & Lite e PSP Go

Actualmente quase qualquer aparelho electrónico permite jogar pelo menos jogos básicos, desde os telemóveis aos leitores de livros electrónicos, passando por leitores de áudio ou vídeo ou receptores de GPS. Mesmo em situações em que os jogos não são fornecidos como parte das funcionalidades de um produto, *hackers* em todo o mundo encarregam-se muitas vezes de transformar aparelhos de forma a permitir a utilização de jogos. A título de exemplo, a maior parte das calculadoras Canon apresentam um *Easter Egg* (uma surpresa escondida) – permitem jogar o jogo Pong com a introdução de um código: 332652 seguido das teclas “+” e “-” em simultâneo.

Além dos dispositivos electrónicos que oferecem jogos nativamente – embora não seja essa a sua função principal – e daqueles que não os oferecem mas por acção de interessados passam a disponibilizar alguns jogos, o conceito de consola de jogo está a ser complementado de várias formas. Num híbrido de jogo tradicional com a electrónica, o *Geocaching* continua a ganhar adeptos. Este desporto consiste na localização de pequenos “tesouros” escondidos em qualquer parte do mundo com a ajuda de um receptor de GPS e combina o desporto de ar livre com a tecnologia.

Pelas mãos do *Multitouch Barcelona* chega uma instalação baseada no antigo jogo Space Invaders. Em quase tudo idêntico ao original, incluindo nos gráficos pixelizados em poucas cores, o jogo é aqui alterado de forma a que os invasores sejam apresentados numa enorme projecção. A função de eliminar cada um dos inimigos não passa agora pela utilização de uma nave controlada

por um *joystick*, mas sim por uma experiência colectiva de atirar fisicamente bolas às naves inimigas (MultitouchBarcelona).



Fig. 39 – Multitouch Space Invaders no festival OFFF 09

3. Estado da Arte

Como foi referido, este documento centra-se na criação de jogos para equipamentos móveis para os quais existe a possibilidade de desenvolvimento e distribuição gratuita ou de baixo custo. O maior obstáculo ao desenvolvimento de aplicações como *hobby* são os altos valores das licenças cobrados pelas empresas que produzem as consolas. O primeiro caso de estudo de estado da arte será portanto o dos equipamentos que permitam o desenvolvimento com custos baixos ou nulos.

Na criação de um jogo surgem questões e problemas que devem ser experimentados e resolvidos constantemente. Durante a produção de um jogo são elaborados múltiplos protótipos, cada um com um fim específico, para resolver problemas e afinar parâmetros. Neste capítulo é efectuado um levantamento das ferramentas que permitem, de forma rápida e eficaz, efectuar a prototipagem de níveis ou áreas. Serão também analisadas as linguagens e ferramentas que possibilitam a criação e compilação de um jogo completo.

Muitos são os factores que tornam um jogo agradável e que fazem o utilizador voltar repetidamente a jogar. Desde o início dos videojogos que novos métodos foram sendo introduzidos de forma a inovar e cativar a atenção. Estes métodos serão o centro deste estudo.

3.1 Equipamentos

Três tipos de equipamentos são analisados por serem considerados os mais relevantes: Telemóveis, PocketPCs (equipamentos com Windows Mobile, com ou sem comunicação GSM/UMTS) e consolas portáteis. Os telemóveis são utilizados diariamente por quase toda a gente e acompanham as pessoas quase sempre. São uma referência devido à quantidade de equipamentos disponíveis e à sua abrangência. Os dispositivos com Windows Mobile apresentam algumas vantagens específicas, como a possibilidade de ecrã tátil (normalmente resistivo, por oposição à camada sensível ao toque capacitiva de equipamentos como o iPhone) e o desenvolvimento sobre .Net Compact Framework. Por ser um fenómeno de vendas e pela forma como proliferam as aplicações no AppStore, o iPhone será mencionado. Não será, no entanto, alvo do produto deste estudo devido à forma como é limitado o acesso e a instalação de aplicações no equipamento. As consolas, por seu lado, são de menção obrigatória devido à qualidade da experiência conseguida. Os controlos, o ecrã, os componentes e a ergonomia são optimizados para fornecer a melhor experiência de jogo possível.

Dentro dos telemóveis, as gamas mais baixas apresentam algumas dificuldades por não permitirem a execução de aplicações, à excepção daquelas que o telemóvel traz de origem. Neste momento, no entanto, é possível adquirir telemóveis que utilizam a tecnologia Java por um baixo custo. Tal como Java, alguns dispositivos permitem a utilização de jogos desenvolvidos com Mophun ou ExEn.

A Nokia, sendo o fabricante de telemóveis com mais modelos vendidos, é uma marca de referência. O Sistema Operativo Symbian, desenvolvido inicialmente em 1998 pela Symbian Ltd,

constituída pela Ericsson, Nokia, Motorola e Psion, é encontrado na maior parte dos *Smartphones*. Em Junho de 2008 a Nokia anunciou a compra de 52% das acções da Symbian Ltd (Deffree 2008), sendo neste momento a detentora do Sistema Operativo. Como resultado desta aquisição são esperados preços mais baixos em telemóveis de gama média/alta, permitindo a utilização destes telemóveis por um público mais alargado. Existem dezenas de milhares de aplicações para Symbian, sendo uma grande parte delas jogos, incluindo aqueles que fazem parte da marca N-Gage, da própria Nokia. Alguns modelos de telemóveis, como o Nokia N95, têm aceleração gráfica.



Fig. 40 – Telemóvel Nokia N95

Os PocketPCs são assistentes pessoais digitais com ou sem funções de comunicação, que utilizam o Sistema Operativo para este tipo de equipamentos desenvolvido pela Microsoft. Tendo como base o Windows CE, este Sistema Operativo apresenta parecenças com o Windows XP e oferece um largo leque de capacidades. Devido à facilidade de desenvolvimento, ao suporte da Microsoft e à grande comunidade de utilizadores, o número de aplicações disponíveis para Windows Mobile é extremamente elevado. Os dispositivos actuais apresentam regularmente ecrãs a cores com resoluções entre QVGA (320x240) e WVGA (800x480), som estéreo de alta definição, teclados fixos ou deslizantes, vários tipos de comunicações sem fios (Infravermelhos, GSM, UMTS, HSDPA, *Bluetooth*, *Wifi*), acelerómetros e elevadas capacidades de armazenamento que podem ser ampliadas com a utilização de cartões SD (incluindo nas versões mini e micro). O ecrã táctil da gama Windows Mobile Professional ou Classic ajuda à experiência de jogo e abre inúmeras possibilidades, assim como a utilização de acelerómetros.



Fig. 41 – PocketPC Qtek S100

O iPhone da Apple, assim como o iPod Touch, utiliza acelerómetros e um ecrã sensível ao toque por meio de uma sistema capacitivo. Este tipo de sistema não é tão preciso como os resistivos utilizados normalmente nos PocketPCs, mas permite *multitouch* e não é preciso fazer pressão para esta funcionar. Não é, no entanto, utilizável com canetas e estiletes. As capacidades gráficas do iPhone são bastante boas para um dispositivo deste tipo, visto que o terminal está munido de aceleração gráfica fornecida por um co-processador gráfico *PowerVR*. Os mecanismos de entrada de dados permitem jogos originais, mas não são práticos para jogar os jogos tradicionais. O iControlPad vem ajudar a combater esse problema. Vindo de uma equipa de aficionados, este dispositivo adiciona controlos de jogo ao iPhone. Ainda não está à venda e está em fase de testes, mas tem por trás uma grande equipa de programadores como ZoddTTD, responsável por projectos como os emuladores de Playstation ou Gameboy Advance. Infelizmente o iPhone é completamente fechado e é necessário recorrer a processos de *JailBreaking* (libertar o iPhone de forma a permitir a instalação de aplicações por meios não suportados pela marca). Por este motivo o iPhone não será considerado, assim como o Android da Google. São no entanto aqui mencionados pela sua inovação, capacidades e pela fama alcançada.



Fig. 42 – Apple iPhone e T-Mobile G1 (Android)

Os terminais até agora referidos são de menção obrigatória devido à sua popularidade e disponibilidade. Nos últimos anos temos vindo a assistir a uma integração de várias funcionalidades num só equipamento, e uma das funcionalidades dos equipamentos até agora expostos é realmente a execução de videojogos. Apesar da inegável vantagem de acrescentar potencialidades a um dispositivo, a melhor experiência obtém-se normalmente quando o *hardware* é desenhado para uma função específica. As consolas portáteis são desenhadas para jogar jogos, tendo a ergonomia estudada para tal fim e dispondo de meios de entrada e saída de dados adaptados a essa função.

Estas consolas serão referidas mais pormenorizadamente do que outros equipamentos, e os protótipos propostos terão como alvo a consola *GP2X F200*, embora possam funcionar em outros dispositivos com poucas ou nenhuma alteração.

A Gamepark e posteriormente a Gamepark Holdings foram as principais responsáveis pelo desenvolvimento e evolução das consolas portáteis abertas. A primeira consola criada pela GamePark foi a GP32 FLU (*Front-lit unit*) que mereceu melhoramentos até 2004, ano em que foi lançado o modelo BLU (*Back-lit Unit*). Vendida a um preço de €199, não apresentava nesta área qualquer vantagem em relação às conhecidas Sony PSP ou Nintendo GameBoy Advance/DS. O público-alvo era, no entanto, consideravelmente diferente. Apenas vinte e oito jogos comerciais foram lançados, mas a verdadeira utilização desta consola – assim como viria a ser a das suas sucessoras – era a execução de jogos *homebrew* (desenvolvidos em casa) e emuladores de sistemas mais antigos. A GP32 foi uma consola de culto entre os seguidores da *scene* (adeptos e entusiastas que apoiavam o sistema com *hardware* e *software* desenvolvido em casa) e representou o início de uma época (Grand et al. 2004).



Fig. 43 – Gamepark GP32

Após o sucesso relativo da GP32, os elementos da empresa Gamepark dividiram-se. Uma parte ficou na Gamepark e dedicou-se à criação de uma consola equiparável, em termos de poder de processamento e renderização de gráficos 3D, à Sony PSP - a XGP - consola esta que nunca chegou a ser comercializada. Por seu lado, os membros da equipa que fundaram a Gamepark Holdings, criaram uma consola com objectivos mais modestos mas cobrindo as necessidades dos fiéis seguidores. Esta consola, de seu nome GP2X (Cantor 2006), foi até recentemente a mais avançada

consola portátil totalmente aberta existente no mercado, apesar das suas relativamente modestas especificações. Nas suas versões F100 e F200 (a F200 com *touchscreen* e ligeiras alterações nos controlos de jogo), esta consola apresenta um processador ARM, uma entrada para cartões SD e utiliza o Sistema Operativo Linux, simplificando o desenvolvimento e distribuição de aplicações. Neste momento a Gamepark Holdings propõe também o modelo F300, em série limitada, apenas para o mercado Coreano. Os principais impulsionadores da *scene* em Inglaterra, Espanha e Alemanha (respectivamente CraigiX, Anarchy e Evil Dragon) estão a analisar a hipótese de importação de algumas unidades. A F300 mantém o formato das suas antecessoras, mas o interior assemelha-se ao da nova consola GP2X Wiz, referida em seguida (GPH 2008).



Fig. 44 – Gamepark Holdings GP2X-F100

A GP2X Wiz é a sucessora da GP2X original, e introduz algumas melhoras e actualizações. O formato de bolso semelhante ao da Gameboy Micro (uma Gameboy Advance mais pequena), aliado ao processador mais rápido – 533Mhz em relação aos 200Mhz da GP2X - e à aceleração 3D incluída oferece a um público mais exigente uma consola actualizada para os tempos que correm. As pilhas da GP2X (uma das características mais criticadas) deram lugar a uma bateria interna de 2000mAh que, segundo indicações da empresa que a criou, permite ao processador ARM9 apresentar filmes ou jogos durante até sete horas, uma grande melhoria em relação à autonomia da antecessora. A Wiz oferece compatibilidade parcial com a GP2X, sendo que a maior parte das aplicações deverão ser recompiladas para que possam ser executadas. Tendo em conta o nível médio de conhecimentos de programação do público-alvo, em pouco tempo toda a biblioteca de software da GP2X deverá estar disponível para a Wiz. Devido ao mais elevado poder de processamento e à aceleração 3D, a Wiz deverá suportar um emulador da Sony Playstation original a ser executado em tempo real, à velocidade da famosa PSOne. Implementações de jogos cujo código foi liberado, como o Quake I ou II deverão apresentar uma execução mais fluida. A GP2X Wiz permite ainda a execução de jogos em Flash, o que aliado ao seu baixo preço promete uma coexistência com a futura consola de topo OpenPandora (GPH 2008).



Fig. 45 – Gamepark Holdings GP2X Wiz

Muitos são os equipamentos de baixo custo provenientes da China que cobrem as mais variáveis gamas de utilização. As consolas de jogos não são excepção, e lojas *on-line* e serviços como o eBay têm vindo a oferecer aos clientes ocidentais um leque de consolas baratas. Baseadas internamente no mesmo hardware e utilizando versões do mesmo Sistema Operativo, as micro-consolas da Gemei, SmartQ ou JXD chegaram a ter vários programadores entusiastas a trabalhar em melhoramentos caseiros e pequenos jogos. Devido à falta de apoio e documentação, aliada ao curto período de vida de cada um destes equipamentos, os esforços destes programadores não chegaram a apresentar resultados significativos. Esta situação está a mudar recentemente com a introdução dos leitores de MP3/MP4/consolas Gemei x760+ e Dingoo A-320. Utilizando o mesmo processador, memória e *firmware*, estas duas consolas diferem quase que exclusivamente na resolução do ecrã, numero de botões e formato. São compatíveis entre si e são programáveis, existindo presentemente uma distribuição de Linux para elas, o Dingux, compilada por seguidores. Estas novas consolas são vendidas a menos de cem euros e oferecem excelentes capacidades por muito baixo preço.



Gemei X720



Fig. 46 – Dingoo A-320

Algumas comunidades foram sendo formadas em volta das consolas abertas, em países como Inglaterra, Alemanha, Espanha, Estados Unidos ou Canadá. Sempre activos, os utilizadores dos fóruns e *blogs* relativos a estes equipamentos sugeriam melhorias, alterações ou novidades que consideravam importantes. Os moderadores de algumas destas comunidades decidiram compilar as melhores dessas ideias e criar uma consola, pensada inicialmente por CraigiX, impulsionador deste tipo de aparelhos em Inglaterra. Combinando o trabalho com o alemão EvilDragon e o canadiano

MWeston, e mais algumas pessoas talentosas nas áreas de *hardware* e *software*, desenvolveram a consola OpenPandora. Esta consola está, à data, em fase de produção, sendo esperada no mercado ainda no decorrer de 2009. Utilizando controlos de jogo analógicos e digitais, um ecrã TFT com uma resolução de 800x480 pixéis, entradas duplas para cartões SD, *Wifi*, *Bluetooth*, saída para televisão, o mais rápido processador ARM da Texas Instruments até à data (Cortex A8, baseado da arquitectura ARMv7) e mais alguns componentes nunca usados neste tipo de dispositivo, a OpenPandora representará, aquando da sua distribuição, o estado da arte nesta área (OpenPandora 2007).



Fig. 47 – Consola OpenPandora

3.2 Ferramentas de desenvolvimento e de prototipagem

Para acelerar o desenvolvimento, várias tecnologias surgiram, para praticamente todos os equipamentos excepto os mais básicos. Cada uma destas tecnologias apresenta particularidades em relação às outras, e nenhuma apresenta só vantagens ou desvantagens. Existirá sempre um compromisso de velocidade de desenvolvimento *versus* velocidade de execução. Ferramentas de mais alto nível permitem rápidos tempos de desenvolvimento, mas podem ser penalizadas em termos de versatilidade ou velocidade de execução.

A prototipagem deve ser conseguida rapidamente, ainda que nem todos os aspectos do jogo estejam totalmente funcionais. O jogo final, por seu lado, deve ser executado sem quebras e todas as funcionalidades devem estar presentes. É normal a utilização de ferramentas de mais alto nível para o desenvolvimento de protótipos, enquanto o jogo final e o seu motor são normalmente desenvolvidos em linguagens de baixo nível. Se o jogo não exigir demasiado, nada impede a utilização de ferramentas de alto nível para o desenvolvimento de um jogo completo. No presente caso, e visto que se tenciona demonstrar a possibilidade de criação de um jogo por indivíduos sem grandes conhecimentos técnicos, é proposta a utilização da ferramenta Game-Editor.

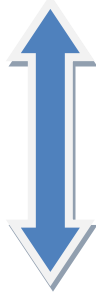
Game-Editor		Alto Nível
Adobe Flash / Flash Lite		
Python / PyGame – Lua / Love - Fenix		
Java / Mophun / ExEn		
.Net Compact Framework (Visual Basic / C#)		
C++ / EdgeLib		Baixo Nível

Tabela 1 – Ferramentas de prototipagem / desenvolvimento

A linguagem Java surgiu em 1995 saída dos laboratórios da empresa Sun Microsystems. A ideia inicial pensada para esta linguagem era a de simplificar a criação de aplicações para praticamente todo o tipo de máquinas. Neste momento, a plataforma Java é muito mais do que isso. É uma linguagem compilada para *bytecode* que por sua vez é executado por uma máquina virtual. Isto significa que os programas criados em Java são portáteis e correm em qualquer equipamento que disponha de uma máquina virtual de Java (JVM ou *Java Virtual Machine*). Esta máquina virtual não passa de um programa que emula um pseudo-computador e transforma, de uma forma extremamente otimizada, todo o código que lhe é fornecido em instruções reconhecidas pelo sistema anfitrião (Byous 2003).

A máquina virtual de Java existe pré-instalada na maior parte dos telemóveis da actualidade, tornando possível o desenvolvimento de aplicações de uma forma mais rápida e numa linguagem orientada a objectos, mais simples do que o C/C++/Assembly. Outra grande vantagem desta linguagem, e talvez a característica que mais contribuiu para a sua implantação é a portabilidade do código intermédio. Como os programas são compilados para correr numa máquina virtual, podem ser executados em telemóveis de qualquer fabricante, em computadores, leitores de multimédia, entre muitos outros. O programador precisa apenas de criar a aplicação uma vez e, desde que tenha tido em consideração algumas diferenças de *hardware* como a resolução do ecrã ou profundidade de cor, esta poderá ser executada em qualquer equipamento. Sendo uma linguagem orientada a objectos, o código pode ser reaproveitado, e funções muitas vezes utilizadas precisam de ser escritas apenas uma vez para que possam ser utilizadas em todos os jogos. Neste momento, a maior parte dos telemóveis suporta a tecnologia Java. A maior parte dos modelos utiliza a versão J2ME (*Java Platform, Micro Edition*) que é uma versão reduzida da plataforma Java. O SDK (*Software Development Kit*) pode ser descarregado gratuitamente e não é necessário o pagamento de licenças para a distribuição ou comercialização de jogos criados com Java, ainda que alguns dispositivos obriguem a que as aplicações sejam assinadas digitalmente.

O Mophun é um motor de jogos para dispositivos portáteis criado pela empresa Synergenix (Pulli et al. 2007). Embora exista pré-instalado em alguns equipamentos sem capacidades de comunicação de voz, como alguns leitores de música/vídeo da Archos, o alvo principal desta

tecnologia é o mercado dos telemóveis. Mais de 20 milhões de equipamentos (MidletReview 2004) trazem Mophun instalado e este é usado pela própria Nokia na produção de alguns jogos da sua linha N-Gage. A API (*Application Programming Interface*) inclui rotinas para acelerar o desenvolvimento de jogos, cobrindo a maior parte das necessidades. Além das funções cobertas pelo Java como desenho de linhas ou rectângulos, o Mophun inclui funcionalidades de detecção de colisão, *sprites*, criação e manutenção de mapas, funções 3D e de comunicação.

A máquina virtual Mophun é complementada em alguns aparelhos por código nativo. A execução dos jogos é bastante rápida, muito mais do que a de jogos baseados em Java. O motor Mophun é extremamente poderoso, mas requer uma licença paga.

A empresa in-fusio desenvolveu o ExEn (*Execution Engine*), com muitas semelhanças com o J2ME. A máquina virtual ExEn apresenta ganhos de performance de até trinta vezes em relação a uma máquina virtual genérica. Vários equipamentos incluem já a tecnologia ExEn, e o futuro apresenta-se promissor para esta tecnologia (Amaro 2003).

Uma das ferramentas de criação de jogos e aplicações lúdicas de mais rápido desenvolvimento é o Flash, da Adobe, e para dispositivos com menor poder de processamento foi desenvolvida a versão Lite (Wayne 2006). Com um ambiente de desenvolvimento que permite a criação directa de gráficos vectoriais, manipulação de texto, som ou mapas de bits, é possível criar jogos com poucas linhas de código. Tal como em Java, a fonte de Flash é compilada para um ficheiro de código intermédio, que será apresentado pelo Flash Player instalado no equipamento. A lógica do jogo é criada em ActionScript, uma linguagem de *scripting* extremamente poderosa, com uma sintaxe relativamente parecida com a do Java. Além destes pontos fortes, a enorme comunidade de programadores e designers que trabalham com Flash pode utilizar o Flash Lite com uma curva de aprendizagem quase inexistente. Com uma forte empresa como a Adobe, tanto o presente como o futuro do Flash Lite enquanto plataforma de jogos para equipamentos móveis se apresentam bastante positivos. Sistemas como PocketPCs e algumas consolas, como as coreanas MyRacer ou a Wiz da Gamepark Holdings, trazem já o Flash Player instalado.

Vindo de um dedicado programador brasileiro, Makslane Rodrigues, o Game-Editor é uma ferramenta de prototipagem e desenvolvimento que permite a criação de jogos para alguns dispositivos, como os PocketPCs e Smartphones da Microsoft, ou a consola GP2X da GamePark Holdings. Tal como o Flash/Flash Lite, um editor WYSIWYG (*What You See is What You Get*) permite posicionar todos os elementos no ecrã. Gráficos, animações ou sons podem ser desenhados visualmente num mapa de jogo. Jogos completos podem ser criados nesta ferramenta de alto nível, sem código algum, sendo que, sempre que necessário, o editor de *scripts* permite adicionar qualquer tipo de funcionalidade. Além dos *scripts*, eventos, colisões, temporizadores e outros elementos podem ser geridos visualmente. Aos jogos criados pode ainda ser adicionado código global em C++ (Rodrigues 2003). O código global é reutilizável, criando novas funcionalidades que posteriormente podem ser aplicadas a qualquer jogo, permitindo a introdução de elementos como caixas de diálogo ao estilo dos jogos RPG ou efeitos gráficos gerados em *runtime*. Os *scripts* criados podem também ser adaptados ou usados sem alterações, de forma a dotar elementos de vários jogos ou dentro do mesmo jogo de comportamentos similares. Esta reutilização de código facilita uma vez mais a criação de jogos de uma forma rápida.

No caso dos PocketPCs, as ferramentas da Microsoft aceleram o desenvolvimento. O .Net Compact Framework pode ser instalado nestes equipamentos permitindo assim o desenvolvimento em Visual Basic ou C#. C++ continua no entanto a ser a linguagem preferencial de forma a conseguir a velocidade necessária para alguns jogos. Motores como o EdgeLib podem ser utilizados para otimizar a criação do jogo. Esta ferramenta tem no entanto um preço relativamente alto, e as licenças começam em €1450+IVA pela versão base (EdgeLib 2005). O Game-Editor, mencionado acima, permite também o desenvolvimento de protótipos e jogos completos para PocketPC.

3.3 Jogos

O jogo “Worm”, criado em 1978 para o *TRS-80* foi publicado no mesmo ano. O jogo e as suas variantes foram publicados para vários sistemas desde então, mas a maior parte das pessoas conhece-o como o “jogo da cobrinha”, dos telemóveis Nokia. O poder de processamento de um telemóvel de gama média é bastante superior ao de computadores existentes na altura, e por esse facto os telemóveis são plataformas ideais para recriar os clássicos. Especialmente nas gamas mais baixas em que todas as aplicações são programadas especificamente para um modelo ou pequena gama de modelos, numa linguagem de baixo nível, os jogos são bastante simples, muitas vezes monocromáticos e mudos ou com sons muito básicos, como os computadores de fins da década de 70/início da década de 80.



Fig. 48 – Tandy TRS-80 IV

Tendo em conta as limitações de *hardware* dos equipamentos alvo, torna-se ainda mais importante a utilização de técnicas para aumentar o interesse, fazer o jogador voltar a jogar após perder ou acabar um jogo ou incentivar a exploração de formas diferentes. Exemplos da evolução das técnicas utilizadas para criar jogos são evidentes em casos como o jogo “Worm”, mencionado no parágrafo anterior. Um jogo que já de si era interessante foi complementado com várias actualizações em termos gráficos, sonoros e principalmente em termos de mecânica de jogo. O recente Snakes Subsonic, da Nokia, não é mais do que uma evolução do “Worm” (ou do “Snakes” que acompanha desde longa data quase todos os terminais da marca), utilizando *power-ups* ou novas possibilidades de movimentação para reviver um jogo tão jogado. Outro exemplo é o “Yoshi’s

Island” da SNES ou NDS, descendente do famoso “Mario”. O jogo “Mario” foi um marco histórico e já na altura utilizava vários conceitos importantes, como uma história, objectivos primários e secundários, o mais famoso “Herói Improvável” e um jogo não linear. “Yoshi’s Island” é uma evolução deste jogo e uma versão mais actual, mantendo os princípios do original e melhorando e actualizando conceitos.

O jogo proposto neste documento baseia-se numa mecânica de *puzzles*/plataformas com problemas a ultrapassar para atingir os objectivos. Neste momento, “Little Big Planet” é tido na generalidade como parte do estado da arte neste tipo de jogo. Este título lançado para a mais recente consola da Sony, a PS3, e mais recentemente para a PSP, apresenta elementos diferenciadores em relação à maior parte dos jogos deste género. Se é complicado reproduzir alguns destes elementos (como a impressionante representação das leis da física) em consolas mais pequenas e menos potente, a generalidade dos parâmetros pode ser aplicada em outros sistemas.



Fig. 49 – Little Big Planet – 2008 Media Molecule, SCE/WW

O mundo de “Little Big Planet” é totalmente abstracto. Apesar do elevado poder de processamento da Playstation 3, os criadores de “Little Big Planet” optaram por fugir a uma representação mais ou menos fiel da realidade. Enquanto a maior parte dos jogos para as consolas de sala apresentam gráficos quase fotorrealistas, neste mundo são misturados bonecos de trapos, cidades constituídas por casas bidimensionais num universo tridimensional e uma noção de profundidade ligeiramente diferente da realidade.

Enquanto jogos de simulação tentam normalmente representar a realidade da forma mais exacta possível (o Microsoft Flight Simulator chega ao ponto de complementar a geometria do seu universo idêntica à do mundo em que vivemos com dados de informações meteorológicas recolhidas em tempo real pela internet), os jogos que adoptam uma aproximação mais abstracta presenteiam o jogador com um universo alternativo. Ambas as aproximações são válidas, principalmente em diferentes tipos de jogos, mas a aproximação abstracta convida-nos a entrar num universo diferente, o que por vezes ajuda à experiência. Os vários elementos de um jogo abstracto são diferentes daquilo que seria esperado. Esta aproximação traz consigo a possibilidade de introdução de elementos originais e permite uma grande liberdade na mecânica de jogo.

A história consegue excelentes resultados ao introduzir o jogador no jogo, tornando a acção mais pessoal. Mesmo jogos de desporto ou *puzzles* beneficiam bastante de uma história, ou pelo menos de um contexto. No caso de “*Little Big Planet*”, um boneco de trapos ficaria fora do contexto sem uma história ou motivação. Seja pela apresentação de algumas imagens, algum texto, ou pela introdução de vídeos nos intervalos da acção (*cutscenes*), o jogador aproxima-se do jogo e tem tendência a identificar-se mais com o personagem que controla se souber onde se enquadra a acção.

Se em alguns jogos como simuladores desportivos ou alguns *puzzles* (dos quais são exemplos os famosos Tetris ou Sudoku), a missão é suficiente para integrar o jogador no jogo, a maior parte dos jogos beneficia de uma motivação extra para jogar o jogo. Essa motivação pode ser indicada pela história ou pode já acontecer durante o jogo.

Os objectivos variam de jogo para jogo. Vencer ou acabar o jogo é muitas vezes o objectivo principal (se bem que em alguns jogos, como os chamados “*God Games*” o jogo é um objectivo em si próprio). Como na vida real, objectivos a longo prazo são mais fáceis de alcançar com objectivos secundários, a médio ou curto prazo.

Em “*Little Big Planet*”, é possível recolher itens escondidos, chaves ou bolhas com pontos. Encontrar estes elementos escondidos permite mais tarde personalizar o personagem. Distribuídos pelo jogo estão vários *puzzles* que se intercalam com a acção principal, mantendo elevados os níveis de interesse.

Para chegar de um ponto ao outro, o jogador pode escolher o seu percurso e a maneira de o ultrapassar. Esta aproximação não linear fornece mais liberdade de acção e é possível jogar o jogo várias vezes de formas diferentes. A longevidade do jogo é ampliada e são permitidos largos níveis de experimentação.

Os *power-ups* são itens distribuídos normalmente pelo jogo, que de alguma forma ampliam as capacidades do personagem. Servem não só como objectivos secundários, mas também como de recompensa por algumas acções mais ousadas. Em “*Little Big Planet*” é possível utilizar um *JetPack* para desafiar a gravidade. Os elementos que permitem mais tarde personalizar o personagem seguem também a filosofia dos *power-ups*.

A simulação da física neste jogo está muito bem conseguida, e a interacção com o mundo de jogo é extremamente realista, apesar de este mundo ser totalmente diferente daquele que conhecemos. Outros factores bastante bem explorados são a interactividade e o jogo em conjunto com vários jogadores. O trabalho colaborativo é recompensado largamente, e a experiência de jogo melhora exponencialmente com o nível de cooperação entre vários jogadores.

4. Desenho de um jogo

A criação de um jogo de computador é normalmente um trabalho de equipa, envolvendo artistas gráficos, animadores, músicos, programadores e outras pessoas de diversas áreas, variando com o estilo e complexidade do jogo. O trabalho é dividido em várias componentes que numa equipa profissional cairiam sobre peritos de diferentes valências.

Como processo complexo que é o desenvolvimento de um jogo, este deve obedecer a várias regras e passar por múltiplas fases antes da sua finalização. Como em qualquer projecto, um planeamento bem elaborado reflecte-se não só na simplificação do trabalho futuro como também num desenvolvimento mais suave e fluído, menos propenso a erros e atrasos derivados de situações não contempladas.

A elaboração de protótipos é um passo essencial na criação de um jogo. Por muito pensado que o jogo esteja, é comum a alteração de parâmetros como velocidades, rotas ou distâncias entre elementos. Para minimizar o tempo de testes, um bom planeamento permite de início uma parametrização de ideias e valores que se irão reflectir numa redução do número e complexidade dos protótipos a realizar.

Tal como um filme, uma música ou um livro, um jogo parte geralmente de uma ideia. Esta ideia pode ser gráfica, mecânica ou de qualquer outra natureza, mas com base na ideia inicial outros elementos devem começar a tomar forma. O processo não tem que ser 100% linear e é possível a mecânica do jogo surgir após a criação de um personagem ou as regras de funcionamento aparecerem após uma bela paisagem que servirá de fundo à acção protagonizada pelo herói.

Um jogo feito em casa contém os mesmos elementos de um jogo comercial e a mesma metodologia é aplicada. A gestão do projecto complementa os diversos passos descritos na tabela 3.

Metodologia de criação de um jogo		
Desenho do jogo	Arte	Programação
Desenvolvimento do conceito	Produção de arte conceptual	Desenvolvimento do motor de jogo
Desenvolvimento da história	Criação de uma identidade visual	Criação de protótipo(s)
Criação dos níveis de jogo	Criação de <i>assets</i>	Implementação da mecânica
Desenvolvimento da mecânica	Modelação	
Animação	Som e Música	Testes
Modelação	Criação e manipulação de efeitos sonoros	Jogo repetido para detectar falhas
Atribuição de "ossos"	Composição da(s) música(s)	
Animações do jogo		
<i>Cut-scenes</i>		
* em jogos 3D ou quando os elementos do jogo 2D têm por base modelos tri-dimensionais		

Tabela 2 – Metodologia de criação de um jogo

4.1 Planeamento

Na fase inicial o papel e o lápis são as ferramentas mais importantes na criação de um jogo. Por muito boa que seja a ideia inicial, esta deve ser trabalhada de forma a corrigir imperfeições, lidar com imprevistos, desenvolver o conceito e aprimorar elementos.

Em todos os jogos actuais, à excepção dos mais simples jogos de *arcade*, uma história é apresentada ao jogador, sendo que o seu desenlace é normalmente ditado pelo desempenho conseguida no jogo. Este passo é de extrema importância porque aproxima o jogador do personagem. É frequente a criação de *storyboards* que auxiliam o criador na tarefa de desenvolver o enredo. Esta história é mais importante em jogos de mecânica linear do que nos jogos de mecânica no estilo *sandbox*, jogos mais livres em que o desenrolar da acção é ditado pelo estilo de jogo do utilizador. Jogos como o famoso “Tetris” ou “Incredible Machine” não têm por trás necessariamente uma história, mas jogos como “Halo” alcançaram a fama devido em grande parte a este factor. Ainda que a história não seja transmitida ao jogador, esta serve para enquadrar a pessoa por trás do desenvolvimento dentro da acção e ajuda a ligar situações ou criar novos elementos.



Fig. 50 – Storyboard – Halo 2

Uma das qualidades inerentes a qualquer jogo, clássico ou digital, é a interactividade. Sem interactividade um jogo não passa de uma exposição. Esta interactividade faz parte da mecânica do jogo e o seu planeamento é um factor chave para o sucesso da produção. Os dispositivos de controlo, como teclados, ratos, *joysticks*, acelerómetros ou sensores biométricos, são considerados e a sua utilização vai reflectir-se numa melhor ou pior experiência final. Na fase de planeamento da mecânica do jogo são estabelecidas regras. Tomando como exemplo o comparativo entre os jogos xadrez e de damas, ambos são jogados num tabuleiro quadriculado com peças de duas cores. As regras mais simples das damas dão origem a um jogo mais simples enquanto os diferentes movimentos das peças no xadrez implicam uma maior concentração e uma estratégia mais elaborada. Este facto não torna o xadrez melhor ou pior do que as damas – cria uma experiência diferente para cada um dos jogos.

A mecânica de um jogo não só dita o seu estilo final como também a experiência resultante e a sua jogabilidade. Com base na mesma história, nos mesmos personagens, no mesmo estilo gráfico ou sonoro, alterando a mecânica de um jogo é possível transformar o produto final de um jogo de plataformas num RPG (*Role Playing Game*).

Após a definição do estilo de jogo, e partindo do exemplo de um jogo de plataformas, vários factores variáveis devem ser especificados. A velocidade dos movimentos do jogador, dos inimigos ou de qualquer elemento interactivo presente no jogo altera também a experiência final, assim como a altura de saltos, a força da gravidade, o tempo que o personagem pode passar debaixo de água ou o número de vidas/quantidade de energia. A criação de protótipos específicos para aprimorar estes factores não dispensa um planeamento nem uma fase de avaliação, mas um bom planeamento pode reduzir o tempo necessário para otimizar estas características.

O personagem principal pode ficar móvel apenas enquanto o jogador carrega numa tecla de movimento, ou pode utilizar inércia e mover-se numa determinada direcção até que outra tecla seja premida. Diferentes tipos de interacção e controlos são definidos antes de o jogo começar a ser criado. O facto de o jogador carregar na tecla que moverá o personagem para a direita (ou mover um *joystick* para a direita) pode significar que o personagem muda automaticamente de direcção ou que a sua componente de movimento na horizontal é incrementada em uma ou várias unidades, alterando a trajectória gradualmente.

O planeamento de um jogo não tem que ser necessariamente um trabalho linear. Um jogo como “Jazz Jackrabbit” poder-se-ia chamar de “*Amazing Weasel*” e manter a jogabilidade com uma doninha em vez de um coelho. Independentemente da mecânica do jogo ser especificada antes ou depois da criação da história ou dos elementos gráficos, todos estes factores aparecem no produto final. A introdução de diferentes personagens é um ponto-chave na criação de um jogo não abstracto. Após a decisão de quem, ou o quê, será o elemento principal do jogo, quais os seus adjuvantes ou inimigos, o ambiente visual pode começar a tomar forma.

Equipas de desenhadores e ilustradores dividem normalmente entre si a criação de elementos. Antes de começar a criar os gráficos finais, sejam eles *sprites* ou modelos tridimensionais com as suas texturas, é necessário um fio condutor que equilibre visualmente a totalidade do jogo. Além dos *storyboards*, a criação de arte conceptual permite transmitir ideias a outros elementos da equipa ou a mantê-la coesa durante toda a produção. As imagens criadas nesta fase são o primeiro passo para dotar o jogo de uma identidade gráfica coerente que permitirá a variedade de mundos e ambientes sem fazer parecer que cada nível representa um produto diferente.



Fig. 51 – Arte Conceptual – Fable 2

4.2 Documentos de apoio

De forma a estabelecer as funcionalidades e regras a que o jogo obedece, são criados documentos que definem as várias componentes do jogo. Juntamente com o cronograma que apresenta prazos para a finalização do trabalho de cada parte da equipa e do trabalho final, outras informações são registadas em documentos como o *Game Design Document*, ou GDD, e o *Level Design Document*, ou LDD. Estes documentos servem como ponto pivô para toda a equipa e permitem o esclarecimento de dúvidas e questões em qualquer ponto do projecto. Em muitos casos existem também documentos mais específicos, como o Guia Artístico, o *Art Asset Document*, o *Sound Design Document* e registos específicos para cada parte da equipa envolvida no projecto (Rodrigues 2009).

Num jogo desenvolvido por uma única pessoa, os documentos escritos não são necessários para a sincronização da equipa, mas não deixam de ser uma ferramenta importante para consulta e ajudam a manter o jogo consistente. Ao fim de alguns meses ou mesmo semanas, é fácil perder o fio condutor do projecto, e a consulta dos documentos permite voltar ao desenvolvimento de novas ideias consistente com todas as linhas guia do projecto. No caso em estudo, dois documentos serão desenvolvidos: o GDD, com informações gerais sobre o jogo, e o LDD, com as regras do nível desenvolvido para o protótipo.

4.2.1 Game Design Document

Embora não exista um formato universal para o *Game Design Document*, originando diferentes modelos em diferentes empresas, o documento deve conter o mesmo tipo de informações, independentemente dos seus criadores ou do tipo de jogo (Ryan 1999).

Para o *designer* do jogo, principal responsável pela elaboração do documento, o GDD tem o intuito de comunicar as ideias e conseguir o trabalho. Serve também como imagem do artigo a

vender para o produtor do jogo. Outro dos responsáveis pela criação do GDD é o director técnico ou um programador sénior, encarregues da definição dos parâmetros técnicos. Para a equipa de programadores o documento serve como guia, de forma a possibilitar o desenvolvimento da mecânica do jogo conforme especificado. Para a equipa responsável pela componente artística, o GDD define o aspecto geral do jogo, assim como as suas componentes específicas. O GDD é normalmente o resultado da colaboração multi-disciplinar entre as várias equipas e pode sofrer alterações ao longo do processo de criação do jogo (Ryan 1999).

O *Game Concept Document* pode servir como documento autónomo ou como subsecção do *Game Design Document*. É um documento de uma ou duas páginas focado no conceito do jogo, e serve como ponto de partida e de união entre equipas para a elaboração ou continuação do GDD. Na generalidade, o GDD é constituído por vários sub-documentos que podem ser consultados independentemente ou como um todo (Ryan 1999).

Iniciando com uma página de rosto que contém dados sobre o jogo e a empresa, o GDD contém informações de diferentes áreas. Para a equipa de *marketing*, informações como o conceito, a sinopse, o mercado alvo ou as plataformas em que o jogo correrá são informações importantes que aparecem logo no início do documento. A equipa artística tira informações de várias secções ou sub-documentos – de um lado recebe informações sobre a aparência visual do jogo, gama de cores ou iluminação, enquanto noutra área pode esclarecer questões sobre *sprites* e animações (Thompson, Berbank-Green, and Cusworth 2007).

Questões relacionadas com a generalidade do jogo aparecem numa secção com a indicação dos objectivos, mecânica, sistema de menus, controlos, visualização de informação durante o jogo, graus e tipos de movimento dos elementos interactivos e não interactivos, sistema de pontuação ou vidas/energia do jogador. Em outras secções do mesmo tipo podem ser encontradas referências à estrutura dos níveis dentro do mundo do jogo, física do jogo, lista de NPCs - *non playing characters* - e as suas características, equipamento e os seus efeitos. O guião do jogo aparece também numa secção do mesmo género, acompanhada com as instruções para finalizar o jogo enquanto jogador (Thompson, Berbank-Green, and Cusworth 2007; Ryan 1999).

Sistemas de câmaras ou movimentos aparecem numa secção técnica, enquanto outra secção deste tipo pode conter informações sobre o motor de jogo, sistemas de colisão, física, tipo de renderização, inteligência artificial, entre outras. Informações relacionadas com o sistema de som utilizado aparecem também nesta secção, com características relacionadas directamente com os sons e efeitos, o diálogo ou música (Thompson, Berbank-Green, and Cusworth 2007).

Outras informações relevantes para o jogo, como a lista de revisões que este sofre ao longo do tempo de desenvolvimento, aparecem finalmente em anexos (Thompson, Berbank-Green, and Cusworth 2007).

O GDD é necessariamente, devido à quantidade e diversidade de informação que contém, um documento bastante extenso. A sua escrita deve no entanto obedecer a parâmetros que facilitem a sua leitura e a procura de informação. O texto deve ser preciso, sem deixar espaço a ambiguidades, e conciso para simplificar a pesquisa e análise (Thompson, Berbank-Green, and Cusworth 2007).

4.2.2 Level Design Document

Enquanto o GDD é um documento geral relativo a todo o jogo, os LDDs existem em quantidade igual ao número de níveis que fazem parte do jogo. Estes LDDs podem ser incluídos como subsecções do GDD. No *Level Design Document* o designer do jogo especifica os pormenores de cada nível, ligando-o sempre que necessário ao nível anterior e/ou posterior ou *cut-scenes* (Thompson, Berbank-Green, and Cusworth 2007).

O LDD é o documento que vai permitir a todas as equipas elaborar os níveis, com especial relevo para a equipa de programadores que vai montar o nível dentro do jogo. Todas as indicações relevantes devem estar presentes e facilmente assimiláveis a partir do documento. Ao longo do tempo de criação do jogo, o LDD é consultado inúmeras vezes por várias pessoas e, como tal, deve ser mantido simples e conciso de forma a poupar tempo a quem necessite de algum esclarecimento. É também importante que seja tão fácil quanto possível encontrar a informação necessária, pelo que as ideias devem estar marcadas e fáceis de encontrar (Thompson, Berbank-Green, and Cusworth 2007).

O nível a que diz respeito e a parte do jogo em que este se enquadra estão presentes no início do documento, assim como o tempo estimado para a conclusão do nível em questão. Sempre que relevante é incluído um mapa e, em jogos que apresentam uma mecânica linear, o caminho que o jogador deve seguir, complementado com os objectivos e sub-objectivos a atingir. Na descrição do nível, e quando os jogos obedecem a guiões pré-definidos, alusões ao guião e alguns excertos são incluídos no documento de forma a ligar a acção à história sobre a qual o jogo é construído. Qualquer acção necessária da parte do jogador é incluída no texto, de forma linear se a mecânica do jogo assim o exigir, ou não linear quando o documento diz respeito a jogos do tipo *sandbox* (Thompson, Berbank-Green, and Cusworth 2007).

No fim do documento é indicada a sequência seguinte que permite a ligação entre dois níveis. Esta pode representar uma animação, um ecrã resumo do resultado, de apresentação do próximo nível ou qualquer tipo de solução utilizada para a ligação entre níveis (Thompson, Berbank-Green, and Cusworth 2007).

4.3 Criação de conteúdos

Numa equipa pluridisciplinar os elementos visuais e sonoros são distribuídos pelos artistas de cada especialidade. Diferentes tipos de jogos requerem diferentes tipos de conteúdos. Um jogo 3D necessita de modelos 3D do mundo em que a acção se desenvolve e dos personagens intervenientes enquanto jogos bidimensionais utilizam imagens e *sprites* para representar os seus componentes. Os conteúdos criados e mencionados neste documento centram-se nos gráficos bidimensionais utilizados no estilo de jogo aqui descrito.

Num jogo bidimensional as representações de todos os objectos são gráficos com duas dimensões que utilizam quase sempre um de três tipos de visualização. Jogos de plataformas ou *puzzles* são normalmente vistos de lado e o mundo em que o jogo ocorre é uma planificação

semelhante a uma fotografia tirada na horizontal. Alguns jogos de carros, desporto ou RPGs (*Role Playing Games*) utilizam uma vista de cima, em que a planificação do mundo e dos seus elementos se assemelha a uma fotografia aérea. Só recentemente é que o poder de processamento de computadores e consolas permitiu a visualização de gráficos tridimensionais realistas, mas desde a infância das consolas e máquinas de *arcade* que os criadores de jogos tentam incluir nos seus produtos a noção da terceira dimensão. Jogos como *Tempest* (clássico da Atari de 1980) apresentam gráficos em três dimensões mas com representações extremamente simples e pouco reais. Vários artefactos foram sendo introduzidos para transmitir a ideia de profundidade ao longo do tempo. A utilização da perspectiva axonométrica isométrica é talvez uma das mais usadas, sendo responsável por inúmeros jogos, especialmente durante a época das consolas de 8 e 16 bits. Este tipo de projecção utiliza medidas iguais para representar a unidade nos três eixos e ângulos constantes de 120° entre estes. Numa representação isométrica de um cubo, todas as arestas apresentam o mesmo comprimento, tal como numa vista superior ou frontal. Esta particularidade permite simplificar a criação de conteúdos pseudo-tridimensionais (Cunha 1999).



Fig. 52 – Vista Lateral



Fig. 53 – Vista Superior



Fig. 54 – Perspectiva isométrica

Devido à pouca memória disponível nos computadores e consolas de outrora, as imagens que servem de fundo à acção são normalmente constituídas por *tiles*. Numa aproximação “*tile based*” os fundos são divididos em partes e montados posteriormente. Utilizando imagens quadradas é criado um padrão, evitando assim a repetição de informação em todos os pontos do ecrã. O resultado é similar a uma montagem de azulejos (*tiles*, em Inglês), em que vários padrões são criados utilizando azulejos com motivos repetidos.

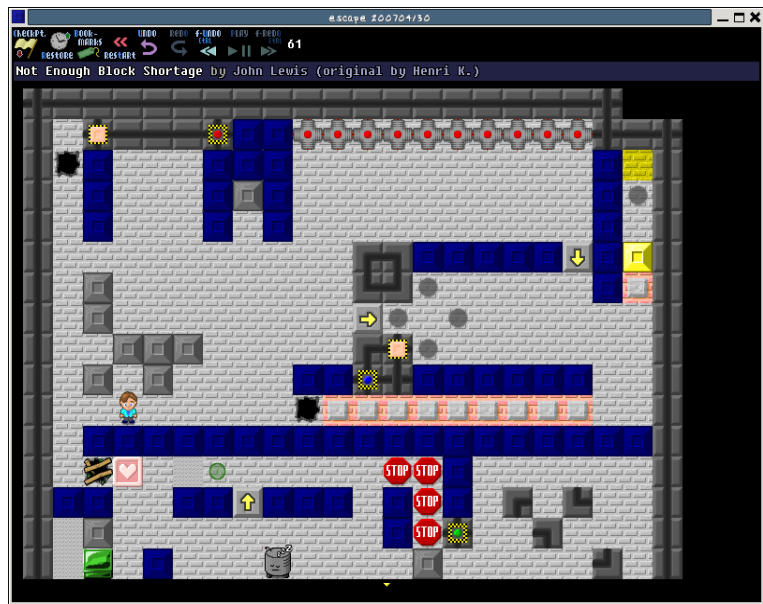


Fig. 55 – Jogo baseado em “tiles”

Num jogo baseado em “tiles” é necessário o planeamento dos níveis tendo em conta o facto de que estes serão criados com base nestas pequenas imagens.

Consolas de gerações anteriores com poder de processamento limitado utilizavam normalmente processadores dedicados à gestão de alguns elementos animados. Estes processadores aceleravam funções como a apresentação e animação de objectos ou testes de colisões. Os gráficos e animações utilizados na maior parte dos jogos bidimensionais herdaram dessa época o nome de *sprites*. É também comum num motor de jogo programado numa linguagem orientada a objectos existir uma classe, geralmente com o nome *Sprite* ou variações, que simula todas as acções originalmente desempenhadas pelos processadores de *sprites* das antigas consolas.

Enquanto a programação do motor de jogos, incluindo o controlo de colisões, fica geralmente entregue a uma equipa de programadores, as imagens e animações constituintes das *sprites* ficam a cargo de artistas gráficos. A criação de *sprites* envolve geralmente a edição gráfica ao nível do *pixel*, porque os pequenos blocos têm que manter o máximo nível de detalhe em mapas de *bits* de tamanho reduzido. São comuns resoluções de 16x16 ou 32x32, semelhantes aos ícones utilizados nos modernos Sistemas Operativos.

Uma das características usuais nas *sprites* é a animação daquilo que representam. Ao contrário dos jogos 3D que utilizam séries de valores numéricos para movimentar um modelo, a animação nos jogos bidimensionais utiliza um método mais semelhante ao vídeo, em que várias imagens com ligeiras alterações são apresentadas em movimento rápido, dando a ilusão de um movimento contínuo. A criação de uma *sprite* implica o desenho de várias posições para o mesmo motivo, normalmente guardado numa matriz uni ou bidimensional. Esta matriz é guardada como várias pequenas imagens ou como uma só, constituída por todas as posições da animação – a folha de *sprites* ou “*Sprite Sheet*” (Gravelyn 2007).



Fig. 56 – Sprite Sheet

Outros elementos gráficos existentes nos jogos incluem fontes (normalmente baseadas em mapas de *bits* e não em coordenadas vectoriais), imagens entre níveis, *splash screens* (imagens que aparecem normalmente no início do jogo com o intuito de transmitir a sua ideia numa imagem), animações ou filmes que completam a história.

A introdução do som nos jogos digitais representou um importante passo na jogabilidade. O som complementa o *feedback* necessário e ajuda a comunicar ao jogador o resultado das suas acções. Quando o jogador perde uma vida ou apanha um item de bónus, o som torna a acção mais real e com mais impacto. É mais fácil para o utilizador ter a exacta noção de um determinado acontecimento quando este é acompanhado de um som. Pequenos elementos de áudio como o tic-tac de um relógio aumentam também a ansiedade transmitida, criando uma imersão mais credível.

A música acompanha os jogos desde o tempo dos osciladores monofónicos das primeiras consolas capazes de emitir som. Actualmente os jogos podem utilizar música polifónica em formato MIDI (*Musical Instrument Digital Interface*), em que cada nota é sintetizada ou reproduzida a partir de uma gravação digital e apenas os valores que representam as notas – com as suas características incluindo altura, intensidade ou duração – são guardados. Esta aproximação traduz-se numa economia de espaço necessária em equipamentos de recursos modestos, como telemóveis, à custa de uma reprodução menos natural e de maiores exigências a nível de processamento ou de componentes dedicados à reprodução de faixas MIDI. Alternativamente, a música pode ser armazenada em formatos análogos ao de um CD ou MP3, mantendo a interpretação original, à custa de espaço de armazenamento (principalmente em formatos RAW) ou processamento (em formatos comprimidos como o MP3) (Ribeiro 2004).

Além dos elementos gráficos e sonoros, outros conteúdos são necessários para a criação de um jogo. Como foi dito anteriormente um jogo partilha certos elementos com um filme. Normalmente existe uma história, personagens, heróis e anti-heróis. Vários objectos são interactivos e é normal a existência de dezenas ou mesmo centenas de personagens com as quais é possível

trocar objectos ou informações, especialmente em jogos do estilo RPG. Num passo fortemente ligado ao desenvolvimento da história, estes diálogos são criados e desenvolvidos.

A criação dos níveis é uma das funções mais importantes, visto que em algumas situações um jogo pode proporcionar uma experiência fantástica ou terrível dependendo do desenho dos níveis a ultrapassar. Jogos como “Sokoban” existem em múltiplas versões, muitas delas com gráficos extremamente básicos e efeitos sonoros simples, mas continuam a ser jogados devido ao desenho dos níveis interessante.

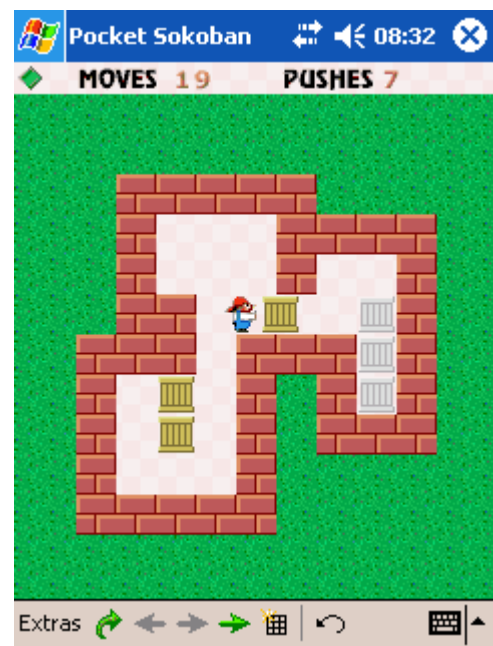


Fig. 57 – Pocket Sokoban

4.4 Programação

Um programa de computador integra todos os elementos criados pelas várias equipas envolvidas na sua elaboração de forma a providenciar a experiência de jogo. Este programa pode ser desenvolvido de raiz para plataformas específicas ou utilizar recursos pré-programados, automatizando parte do processo. Muitas funcionalidades são comuns a vários jogos e em grande parte das vezes é desnecessário programar novamente todas as rotinas.

Actualmente o mercado é extremamente competitivo e os tempos de desenvolvimentos são curtos, pelo que a maior parte das empresas do ramo utilizam motores de jogo proprietários ou livremente acessíveis, diminuindo os tempos e custos de produção. Motores (ou *engines*) 3D (como o OGRE - *Object-Oriented Graphics Rendering Engine*) ou 2D (Game Maker, Game-Editor) estão disponíveis para qualquer pessoa. Com algumas destas aplicações é possível criar jogos completos de forma muito rápida, e em alguns casos a utilização de código torna-se mínima ou mesmo nula.

Em algumas situações, como é o caso do programa Blender, um motor de jogos está incluído ou combinado com editores de gráficos em duas ou três dimensões, permitindo ao criador de videojogos a utilização intuitiva de uma ferramenta para criar vários elementos integrantes do seu jogo.

Motores como o *Doom Engine* ou o *Quake Engine*, ambos da id Software, foram feitos inicialmente para desenvolvimento interno. Com os motores criados, vários jogos puderam ser elaborados sem cair na redundância de programação de certas funcionalidades. Com o passar do tempo, o motor do Quake foi sendo desenvolvido e ambos foram eventualmente tornados gratuitos para que qualquer pessoa pudesse criar os seus próprios jogos sem repetir trabalho já feito.



Fig. 58 – Ambiente jogável criado com Quake Engine

A simplicidade disponibilizada pelos motores de jogo gratuitos tem como custo uma quebra de desempenho bastante suportável em computadores actuais que correm jogos com alguns anos (ou que obedecem aos limites de há alguns anos atrás). Quando o jogo a desenvolver vai correr em equipamentos mais modestos, é por vezes necessária a optimização de todos os recursos disponíveis. O melhor aproveitamento do *hardware* é conseguido quando uma linguagem de baixo nível é utilizada na criação do jogo, cobrindo todas as suas necessidades de forma optimizada.

Num equilíbrio entre rapidez de desenvolvimento e velocidade de execução, a linguagem C é muitas vezes utilizada com bibliotecas específicas que permitem acelerar a criação de rotinas necessárias. Disponibilizadas na forma de APIs (*Application Programming Interface*), estes conjuntos de rotinas permitem funcionalidades como criação de som, apresentação de elementos gráficos, controlo de colisões ou rotação de imagens.

Além de mais rápido tempo de desenvolvimento e da simplicidade de utilização, bibliotecas como o SDL (*Simple DirectMedia Layer*) dotam as aplicações de uma portabilidade que de outra forma seria difícil. Apesar de linguagens como o C obedecerem a regras normalizadas e permitirem a recompilação directa na maior parte dos casos, a programação de funcionalidades específicas como

o desenho no ecrã ou a leitura de dispositivos de entrada varia de sistema para sistema. Sem a utilização deste tipo de bibliotecas, um jogo teria que ter muito do seu código reescrito antes de poder ser compilado para diferentes sistemas. O SDL está disponível para, mas não só, Windows, MacOS, BeOS e Linux (utilizado em múltiplas plataformas, entre as quais a maior parte das consolas abertas).

Sendo um tipo específico de programa, a arquitectura de um jogo segue um plano mais ou menos padrão.

- No início do jogo é apresentado o menu.
- Conforme a opção escolhida, é alterado o estado actual.
- Se o estado actual não for “jogo”, as respectivas funcionalidades são executadas.
- Se o estado for “jogo”, é mantido o ciclo de jogo, ou *game loop*. É possível alterar o estado actual durante o jogo, de forma a apresentar opções de pausa ou menus.
- Durante o *game loop* vários testes são executados. Estes testes podem alterar o estado de forma a apresentar animações intermédias, como no caso de o jogador perder uma vida.
- Se o jogo acabar, por chegar a zero o número de vidas disponíveis ou a energia, o programa volta ao menu inicial.

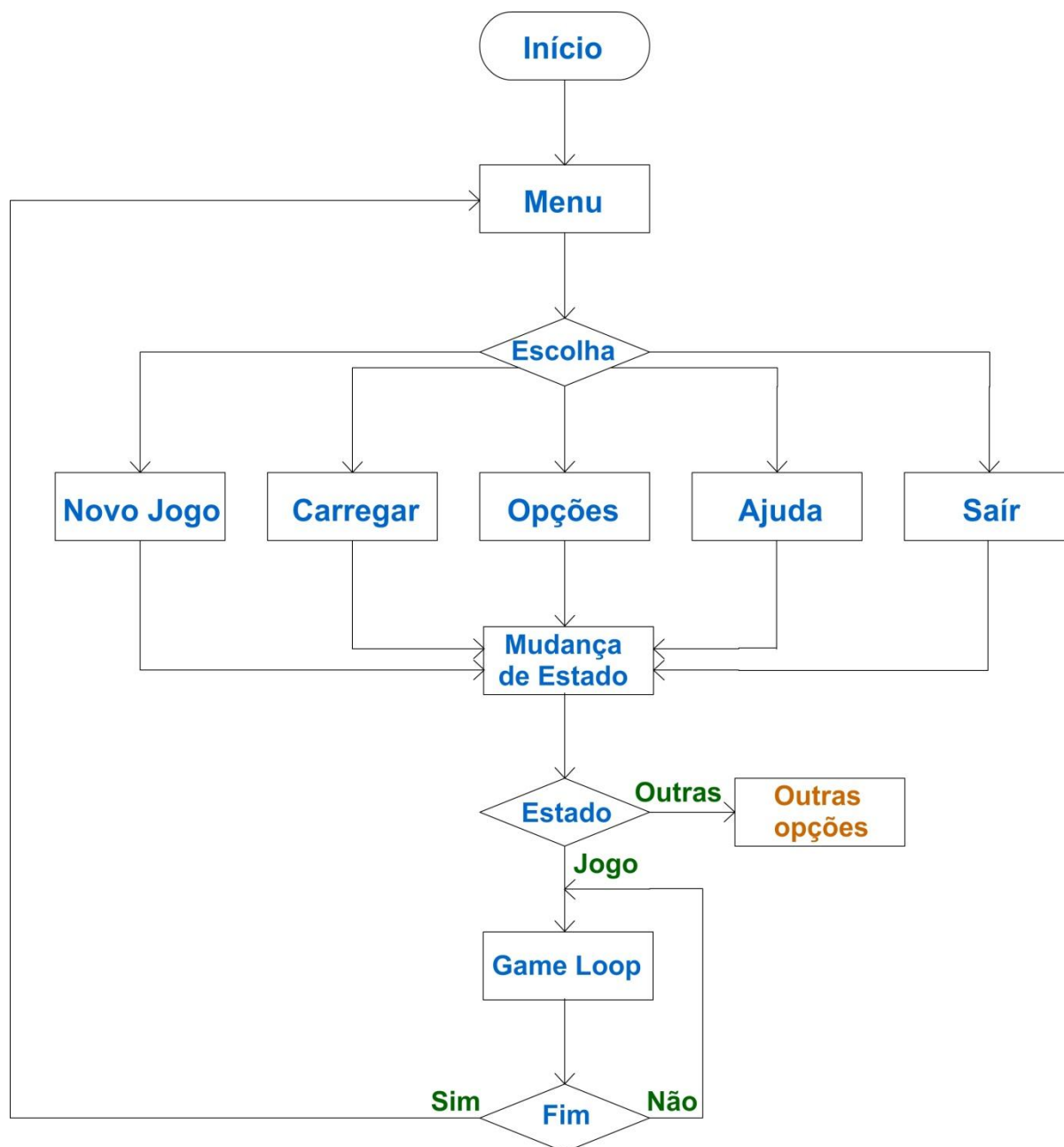


Fig. 59 – Diagrama simplificado da arquitectura de jogo

Dentro desta arquitectura, o *game loop* é a parte que necessita de mais atenção. Devido às características inerentes a um jogo, a acção é constante – ao contrário de outros tipos de programa em que algo acontece apenas quando o utilizador carrega numa tecla ou no rato ou quando chega uma mensagem pela rede – e a interacção provoca resultados imediatos. Por este motivo é necessário o controlo do *frame rate*. Ao contrário do vídeo, o número de imagens por segundo não é a única coisa que deve ser monitorizada. A experiência será positiva apenas se todos os factores forem controlados em paralelo com a animação. Não adianta actualizar os personagens no ecrã sessenta vezes por segundo, se as colisões forem confirmadas apenas a cada dez imagens, ou se a acção do jogador for lida uma ou duas vezes por segundo.

O *game loop* encarrega-se de actualizar constantemente todo o retorno visual e auditivo, confirmar todas as interacções entre elementos do jogo e ler as acções do jogador através do mecanismo de controlo utilizado, seja ele um teclado, *joystick*, rato ou qualquer outro dispositivo.

As duas funções principais do *game loop* são normalmente chamadas de ***update*** e ***draw***.

A função ***update*** trata de todas as decisões de inteligência artificial de que podem ser dotados os personagens do jogo, como perseguir o herói ou evitar paredes. Paralelamente, todas as entradas de dados e novos comandos, seja directamente de um controlador ou a partir de um interface de rede (para jogos com vários utilizadores) são analisadas e os seus resultados calculados. As colisões entre os vários elementos relevantes são testadas, actualizando variáveis ou alterando o estado conforme seja necessário. As posições de todos os objectos móveis são actualizadas conforme as regras preestabelecidas.

Após todos os cálculos executados pela função ***update***, os elementos são visualizados nas suas novas posições pela função ***draw***. As pontuações são actualizadas no ecrã, assim como todos os elementos visuais do jogo. Simultaneamente, os sons relevantes são reproduzidos.

O ciclo conhecido por *game loop* é repetido em regra trinta ou sessenta vezes por segundo até que o estado seja alterado para uma situação em que o jogo não esteja a ser executado. Nesta situação o utilizador é novamente remetido para o menu ou o programa é terminado.

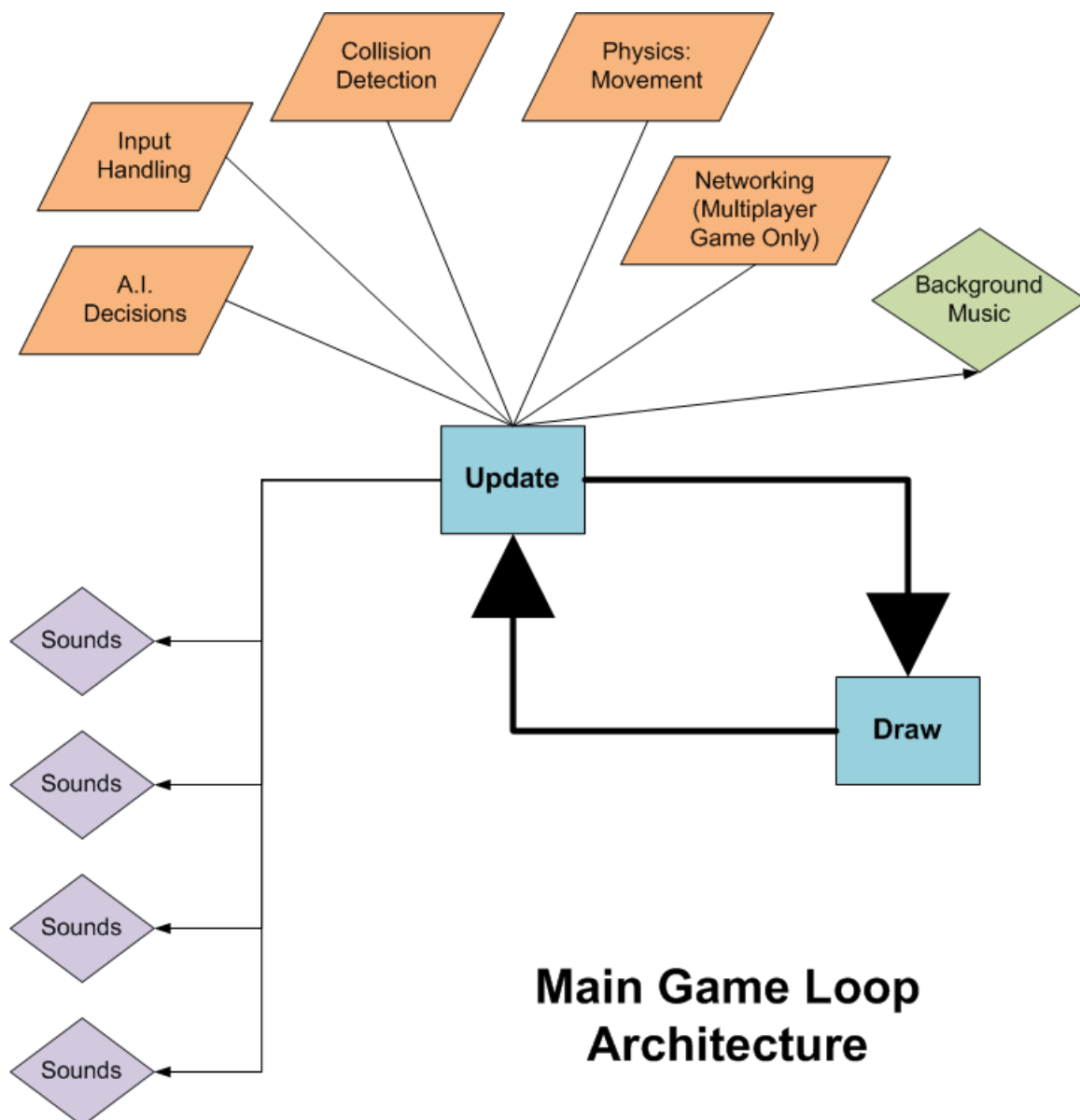


Fig. 60 – Arquitectura do ciclo principal de jogo (CSProjects)

No caso dos jogos de computador, um protótipo é uma pequena aplicação que visa testar conceitos e regras ou experimentar novas situações. Durante todo o desenvolvimento do jogo, são criados protótipos sempre que é preciso experimentar uma situação antes de avançar com o trabalho definitivo.

Uma boa ideia no papel pode não corresponder a um bom produto final. Um protótipo é geralmente desenvolvido após a criação do GDD devido ao facto do custo do GDD ser inferior ao da criação de protótipos. Este protótipo serve como prova do conceito original e, apesar de não apresentar a qualidade do projecto acabado, os conceitos podem ser testados antes de despendere tempo e dinheiro na criação do jogo.

Um bom planeamento minimiza a necessidade de alteração das regras estipuladas para o jogo, mas nem sempre é possível prever na totalidade os efeitos da lógica desenvolvida. A criação de pequenos protótipos pode à primeira vista parecer um adiamento do produto final, mas é mais fácil corrigir pequenas falhas e otimizar pormenores de situações específicas do que ter que fazer alterações de raiz num produto acabado ou quase final. No entanto um protótipo não serve só para testar ideias, serve também para criar ideias, através da experimentação.

Em muitos casos, os protótipos usam gráficos simples, por vezes quadrados ou círculos, para poupar tempo e dinheiro. Esta prática ajuda também a focar a atenção nas questões práticas da mecânica do jogo.

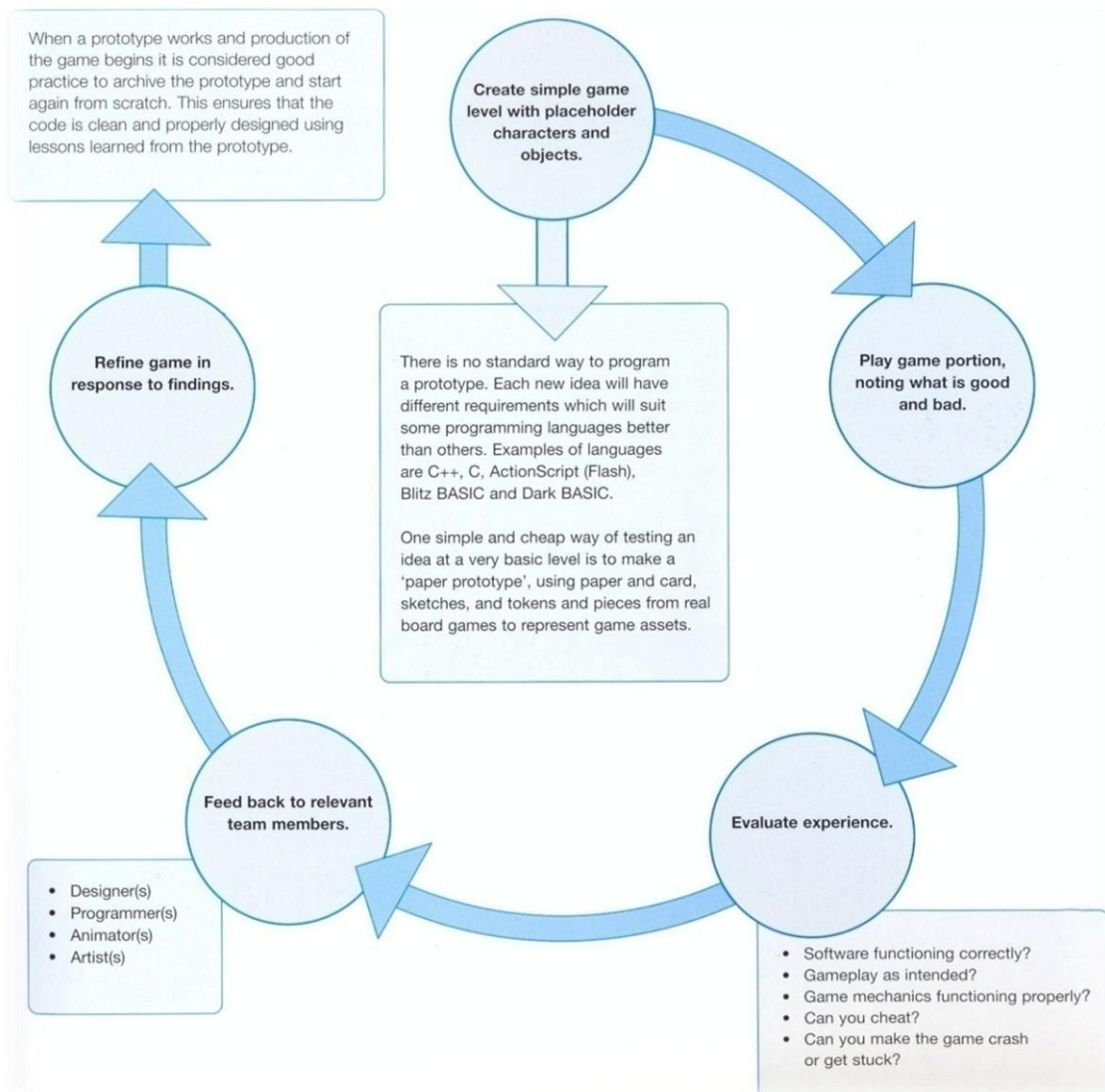


Fig. 61 – Esquema do processo de desenvolvimento e testes (Thompson, Berbank-Green, and Cusworth 2007)

Como normalmente não são incluídos gráficos finais e efeitos especiais nos protótipos, estes podem muitas vezes ser desenvolvidos em linguagens de mais alto nível do que as usadas no

jogo. Os tempos de desenvolvimento dos protótipos devem assim, sempre que possível, ser inferiores ao necessário para incluir uma secção equivalente no produto em produção. Para testar situações relativas à mecânica dos jogos são muitas vezes criados protótipos físicos, utilizando objectos como papel e tesoura.

Outro motivo para a utilização de protótipos é a apresentação do jogo a eventuais patrocinadores. Para convencer uma entidade da viabilidade de um projecto, nada funciona melhor do que a apresentação do mesmo. São por vezes desenvolvidos protótipos – estes sim, com gráficos e efeitos reais do jogo acabado – de um nível ou parte de um nível de jogo, de forma a demonstrar a experiência em vista. Estes protótipos apresentam custos muito mais elevados do que os referidos anteriormente, mas em situação ideal podem trazer uma injeção de capital necessária para a continuação do projecto.

4.5 Testes

Todo o período de desenvolvimento é constantemente acompanhado de testes que validam o trabalho desenvolvido. Os inúmeros protótipos criados servem para testar situações específicas como a jogabilidade ou a possibilidade de um jogador fazer “batota” para ultrapassar uma situação. Estes testes, no entanto, não confirmam que o jogo está pronto para lançamento no mercado.

No final da criação do jogo, este passa por uma fase exaustiva de testes. Testadores profissionais trabalham em empresas de criação de video-jogos e o seu contributo é tão ou mais relevante para o produto final do que o de qualquer outro elemento da equipa. Esta função é o sonho de muitos jovens, especialmente aqueles que não têm os estudos ou capacidades necessárias para fazer parte de uma equipa de *design* ou programação. É também o ponto de entrada nesta indústria para muita gente.

A ideia geral é de que um emprego a testar jogos é algo fácil de fazer e sempre divertido, porque no fundo a função de um jogo acabado é divertir. Especialmente para os mais novos, que nos tempos que correm aproveitam todo o tempo livre para jogar jogos em computadores e consolas, este trabalho pode parecer o sonho de uma vida. A função é tida como aliciante ao ponto de várias páginas e serviços na internet anunciarem os seus préstimos de forma a facilitar a entrada no mundo dos testes de jogos de vídeo em troca de dinheiro. Com uma credibilidade incerta, as frases publicitárias utilizadas variam entre *“Get paid to Play video games! – how I made a killing online of over \$x,xxx.xx per month wasting time playing video games all day...”* e *“What’s your level of education? It doesn’t matter. ANYONE [can] become a video game tester”* e podem ser encontradas em endereços como <http://gametesterguide.net/>, <http://www.only4gamers.com/> ou <http://www.betagamingtesters.com>.

Ao contrário do que é assumido por muitos – especialmente na camada mais jovem – o trabalho de testar um jogo não é fácil, e muitos são os trabalhadores que entram para a função para serem dispensados pouco tempo depois. O trabalho de testar os jogos faz parte do departamento de controlo de qualidade das empresas do ramo. Com salas que podem atingir o tamanho de um

ginásio e acomodar centenas de pessoas, o trabalho é exaustivo e obedece a regras diferentes das utilizadas quando se joga um jogo ao fim do dia, sentado descontraidamente no sofá (Small 2008).

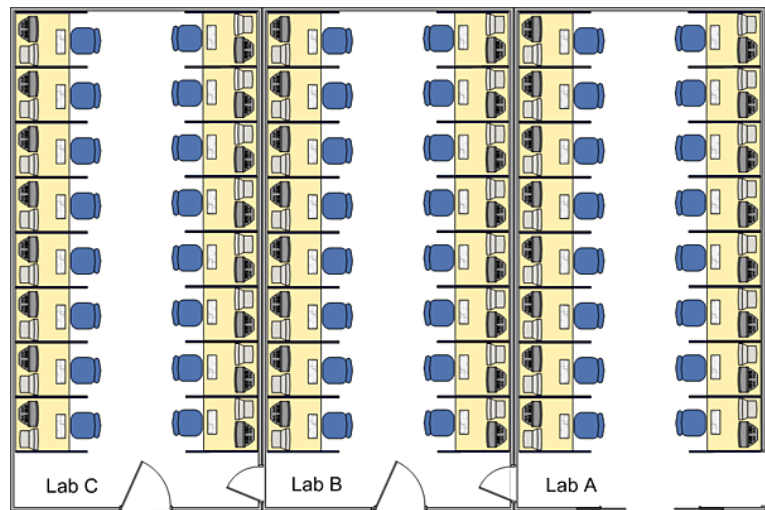


Fig. 62 – Planta de três laboratórios de testes nos Microsoft Game Studio

A cada testador é atribuída uma parte do jogo. Esta parte do jogo é repetida de diferentes formas, tentando encontrar caminhos alternativos para detectar “bugs”. A cada imperfeição encontrada, o testador pesquisa uma base de dados e, se a situação não tiver ainda sido reportada, adiciona uma nova entrada para recomençar o processo. Este trabalho é repetido todos os dias e cada dia de trabalho tem a duração normal de pelo menos oito horas (Sloper 2001).



Fig. 63 – Laboratório de testes dos Microsoft Games Studio

Grandes produtoras têm normalmente o seu departamento de controlo de qualidade que inclui a secção de testes. Em empresas mais pequenas, a mesma pessoa acumula várias funções e é

comum alguém que inicia a sua carreira a testar jogos de computador passar para a equipa de edição ou *marketing*.

Devido à inviabilidade económica de suportar uma equipa com a função de testar jogos, existem no mercado empresas cujo ramo de negócio é exactamente esse: testar jogos de várias editoras. Estas empresas não verificam apenas jogos mas vários tipos de programas (Betabreakers) (Videogametesterhub).

Para um criador de jogos independente, também conhecido como *indie developer*, qualquer das opções tem um custo mais alto do que o suportável. Quando alguém desenvolve um jogo em casa, as fases de testes ficam também por conta do ou dos elementos da equipa. Infelizmente, além de ser mais uma função em cima de uma pequena equipa com pouco tempo e muito trabalho, testes feitos pelos criadores dos jogos são normalmente pouco fiáveis. Existe a tendência a testar os pontos que foram mais complicados no desenvolvimento e deixar de lado situações aleatórias que alguém de fora provavelmente experimentará. Existe sempre uma fase de testes feitos em casa mas que regra geral não é suficiente para assegurar a isenção de erros e problemas.

A solução para os testes de uma produção caseira passa muitas vezes pela distribuição de versões *beta* a um grupo limitado de utilizadores. A publicação de uma mensagem num fórum para pedido de testadores assegura quase sempre um leque alargado de pessoas disponíveis para o trabalho. Se a mensagem for acompanhada de algumas imagens do jogo e uma breve descrição, é praticamente garantida a equipa de testes. Para assegurar que o trabalho de teste vai ser executado, a recompensa de uma cópia do jogo final pelo trabalho serve muitas vezes como pagamento.

5. **Projecto**

O tema em estudo assenta na criação de um jogo para consolas portáteis, sem custos de publicação. Estes custos são o maior obstáculo à criação de jogos “caseiros”, mas recentemente o lançamento de consolas abertas veio contornar esta situação. Com esta ideia em mente, o projecto desenvolvido seguiu os pontos comuns na criação de videojogos adaptando as condições à criação destes jogos por pequenas equipas, de poucas ou uma só pessoa.

Para simplificar e acelerar o desenvolvimento, a ferramenta Game-Editor, de baixo custo e utilização simples, foi utilizada como motor de jogo. Dada a possibilidade de exportação do jogo para diferentes plataformas, o projecto foi desenvolvido de forma a permitir a utilização do mesmo jogo em equipamentos com Windows Mobile.

No caso em estudo, foram utilizadas ferramentas comerciais dada a forma como permitem a criação rápida de conteúdos e por apresentarem um interface e modo de trabalho familiar à maior parte dos programadores, *designers*, ilustradores e demais interessados na área da multimédia e criação de jogos. Estas ferramentas poderão, no entanto, ser substituídas por programas livres e de código aberto.

5.1 **Análise de viabilidade**

As consolas abertas têm vindo a atrair um número crescente de utilizadores, mas este número é muito baixo para justificar o trabalho não remunerado de criação de um jogo (apesar de haver jogos pagos para este tipo de equipamento, a maior parte deles é gratuita e de código aberto, no espírito das próprias consolas). Os telemóveis, por seu lado, embora não apresentem o formato anatomicamente correcto e os controlos ideais para a execução de videojogos, representam uma quota de utilizadores bastante alta.

A utilização do Game-Editor alia à facilidade e rapidez de implementação uma gama de equipamentos relativamente alargada, sem obrigar à duplicação de trabalho. Das plataformas para as quais o Game-Editor permite gerar jogos, os terminais com Windows Mobile apresentam uma maior divulgação e, caso um jogo criado com esta aplicação seja distribuído ao público, estes terminais serão sem dúvida aqueles em que o jogo será mais utilizado.

A gama alta no mercado de telemóveis é dominada pelos Sistemas Operativos Symbian, RIM, Windows Mobile, o Sistema Operativo do iPhone baseado em MacOS X, e recentemente Sistemas Operativos baseados em Linux, como o do OpenMoko ou o Android da Google. Apesar de o Android ter vindo a ganhar terreno, especialmente nos últimos meses, o maior número dos utilizadores continua focado nos primeiros quatro Sistemas Operativos. A RIM (Research in Motion) apresenta vendas bastante altas na América, mas a sua quota é bastante mais baixa noutros mercados, sendo o Symbian, Windows Mobile e iPhone OS os Sistemas Operativos mais utilizados nas gamas altas dos equipamentos de comunicação móveis. O número de utilizadores de Windows Mobile atinge números suficientemente elevados para justificar o desenvolvimento de aplicações para os terminais.

Sistema Operativo	Vendas 2008	Quota de mercado 2008 (%)	Vendas 2007	Quota de mercado 2007 (%)	Crescimento 2007-2008 (%)
Symbian	72,933.5	52.4	77,684.0	63.5	-6.1
Research In Motion	23,149.0	16.6	11,767.7	9.6	96.7
Microsoft Windows Mobile	16,498.1	11.8	14,698.0	12.0	12.2
Mac OS X	11,417.5	8.2	3,302.6	2.7	245.7
Linux	11,262.9	8.1	11,756.7	9.6	-4.2
Palm OS	2,507.2	1.8	1,762.7	1.4	42.2
Outros	1,519.7	1.1	1,344.0	1.1	13.1
Total	139,287.9	100.0	122,315.6	100.0	13.9

Tabela 3 – Crescimento dos Sistemas Operativos para telemóveis 2007-2008, EUA (Gartner 2009)

Ainda com o intuito de analisar a receptividade de um jogo focado em consolas abertas e equipamentos móveis, foi elaborado um inquérito (Anexo 5): 70.5% das respostas recebidas indicam que o inquirido joga videojogos e 43.5% das pessoas que responderam ao inquérito joga jogos em dispositivos móveis. Tendo em conta que dos restantes 56.5% muitos serão os que não jogam jogos em situação alguma, o universo das pessoas que joga em dispositivos móveis apresenta-se animador.

56.7% das pessoas responderam que o preço justo de um jogo para telemóvel seria inferior a €10, sendo que entre essas pessoas 19.4% não estaria disposta a pagar por um jogo. Estes resultados indicam que 80,6% das pessoas considera que um jogo para um dispositivo móvel deve ter um preço. Se a maior parte das respostas coloca o preço justo abaixo dos €10, mais de um quarto considera um valor acessível entre €10 e €20, e 15% consideram justos valores superiores a €20.

Este projecto assenta no desenvolvimento “caseiro” de um jogo, e o facto de as pessoas gostarem de jogar não significa que estejam dispostas a jogar jogos feitos em casa. A qualidade normalmente tem um preço, e muitas das respostas poderão ser provenientes de pessoas que têm como base a ideia de um jogo comercial proveniente de grandes estúdios. No entanto, e perante uma redução de 50% no suposto preço final do jogo, metade das pessoas dispostas a comprar um jogo optaria pela aquisição de um jogo feito em casa.

O objectivo deste trabalho não é de forma alguma a obtenção de receitas com base no produto final, mas os resultados expostos indicam visivelmente que há mercado e interesse no trabalho de criadores *indie* de videojogos.

Ao fim de várias décadas de criação e consumo de jogos de computador, um grande número de ideias para jogos, em termos de visualização, história e mecânica já foram exploradas. Certos conceitos entram para a história e dão lugar a inúmeras versões, cópias ou variações que ocupam os lugares cimeiros nas tabelas de vendas. O exemplo mais evidente será talvez o do jogo “Tetris”, que continua a ser distribuído em diferentes versões, para diferentes plataformas, apesar

de ter completado vinte e cinco anos no corrente ano. Outro exemplo está patente nos *best sellers* de duas das mais conhecidas lojas online de *software* para Windows Mobile: o jogo “Bejeweled 2”, baseado no original “Bejeweled” de 2001. Este jogo ocupa o primeiro lugar nas vendas de jogos na loja Handango (<http://www.handango.com>) e o segundo lugar na Pocketgear (<http://www.pocketgear.com>).

Bestsellers in Games



New in Games



Fig. 64 – Top de vendas de jogos na Handango

Games

Products 1-10 of 3241

1 2 3 4 5 6 7 8 9 10 +1 +10

Sort By: Best Sellers



Advanced Brain Trainer, Edition 2 (PPC)

Windows Mobile PocketPC

★★★★★

With Advanced Brain Trainer Edition 2, enjoy 4 training modules to improve your calculation, logic and reaction time.

Available for Over-the-Air delivery!

\$11.95 USD

ADD TO CART



Bejeweled 2 for Pocket PC

Windows Mobile PocketPC

★★★★★

The legendary, award-winning, color-matching puzzle game for your smartphone!

Available for Over-the-Air delivery!

\$9.95 USD

ADD TO CART

Fig. 65 – Best Sellers na Pocketgear

Outros dois lugares de topo ocupados nas duas lojas referidas pertencem a uma versão de Sudoku e o “Advanced Brain Trainer”. O primeiro aliciou pessoas dos cinco continentes e foi distribuído em todo o tipo de suportes, digitais ou não. O segundo, baseado no “Dr. Kawashima's Brain Training” da Nintendo deu origem a inúmeras cópias e versões para várias plataformas. Este facto demonstra que existe bastante procura para jogos originais e diferentes. Por outro lado, não é fácil desenvolver uma ideia que cativa tão grandes massas de jogadores. Sem querer copiar ou mesmo inovar ideias já utilizadas, uma das metas deste trabalho é precisamente a criação de um jogo diferente e original.

5.2 Planeamento

De tempos a tempos aparece um jogo diferente e viciante pela sua originalidade. O jogo aqui apresentado tem como finalidade a utilização de mecanismos desenvolvidos com o intuito de proporcionar uma experiência diferente e original.

A ideia inicial é exposta no *Game Design Document*, em anexo a este documento. Como resumo, a originalidade do jogo resulta dos seguintes factores:

- A acção desenrola-se em torno de um herói, Godofredo. O jogador não tem, no entanto, qualquer controlo directo sobre este herói.
- Para terminar cada nível, o herói Godofredo deve chegar ao seu destino em segurança, mas o caminho deve ser preparado por um outro herói, este sim, controlado pelo jogador.
- O jogador tem não uma, mas duas preocupações no desenrolar de cada nível. Ambos os heróis devem ser mantidos vivos por interacção do jogador, em vez da mecânica clássica em que toda a atenção é concentrada num só alvo.
- Em vez de um limite de tempo pré-definido para cada nível, existe um limite de tempo para cada obstáculo que depende directamente da velocidade do herói Godofredo. Estes limites podem ser indirectamente modificados com a alteração do trajecto para diferentes terrenos ou apanhando maçãs mas, a cada obstáculo ultrapassado, e devido à velocidade e trajectória à partida constantes para o herói Godofredo, um novo limite é apresentado antes do obstáculo seguinte.

O projecto é complexo e implica muito tempo e trabalho. A intenção não é apresentar um produto acabado, mas sim estudar a possibilidade da criação de um jogo em casa, por uma pessoa ou um grupo de amigos. Por outro lado, a ideia de criar um jogo básico como o Pong não apresenta grandes desafios. A opção seguida é, portanto, a definição dos parâmetros necessários para a criação de um jogo relativamente complexo que sirva como lista de especificações para um desenvolvimento futuro. No final é apresentado um protótipo de um nível do jogo referido, com alguns dos elementos previstos. O protótipo deverá reflectir todas as operações necessárias para a criação do jogo completo e, com base nas ideias e conceitos absorvidos com a criação do protótipo,

um jogo final poderá ser produzido. A diferença entre a criação do protótipo e a criação de um jogo completo será então a quantidade de trabalho a desenvolver e o tempo necessário para completar o projecto.

De forma a definir prazos para a criação dos elementos constituintes do jogo, desde a elaboração do conceito até à fase final de testes, é elaborado um diagrama de *Gantt*. Este tipo de documento constitui uma ferramenta valiosa para o gestor do projecto e, no presente caso, contribui para a organização das várias componentes do trabalho.

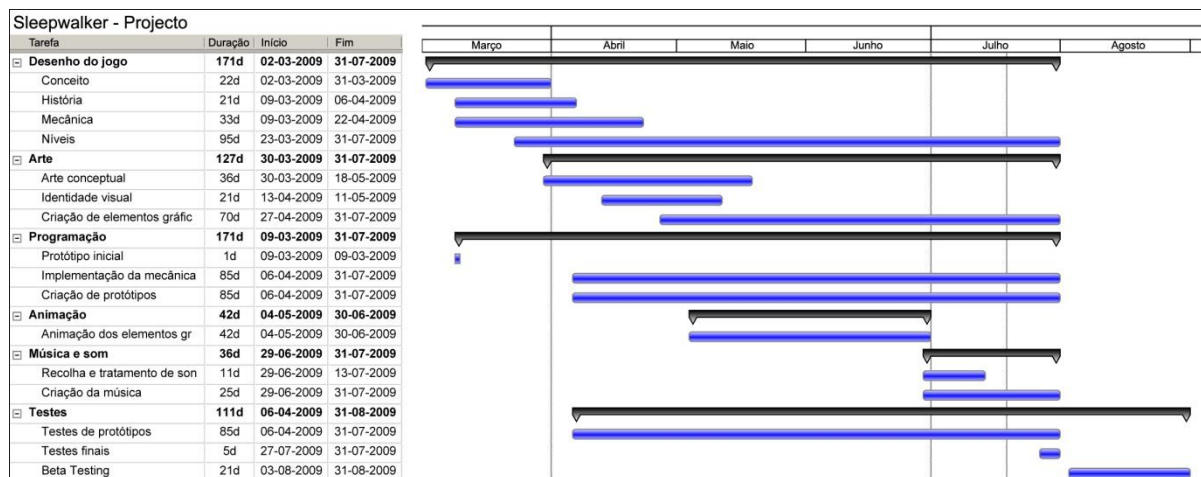


Fig. 66 – Mapa de Gantt – Prazos de desenvolvimento

Partindo da ideia inicial de um personagem controlado pelo jogador a ajudar outro personagem sobre o qual o jogador não tem controlo directo, foi criado o *Game Design Document*. Para testar a viabilidade do projecto, foi desenvolvido um protótipo rápido. Este protótipo inicial, sem elementos gráficos elaborados, provou que a ideia tinha potencial para resultar num jogo com boa aceitação.

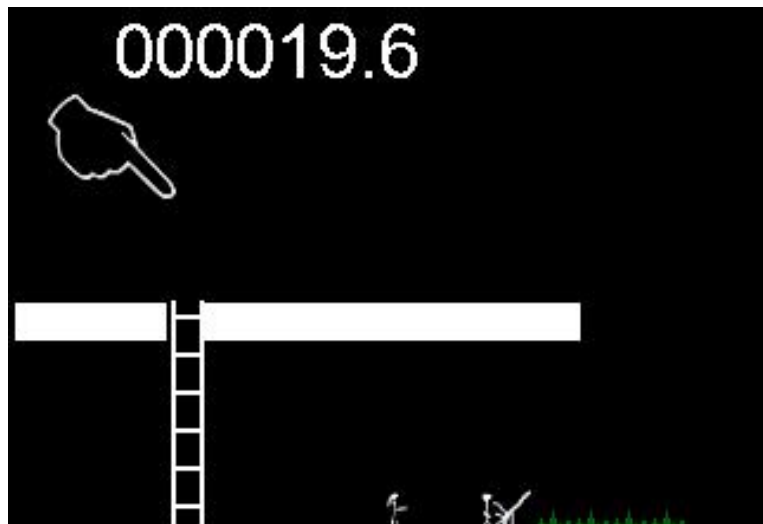


Fig. 67 – Sleepwalker - Protótipo inicial, versão 2

Inicialmente o protótipo centrava a janela de jogo sobre Godofredo. A ideia mostrou potencial nos primeiros momentos, mas quando o herói controlado pelo jogador desapareceu do ecrã, foi confirmado que algo teria que ser alterado. Com a adição de duas “Wire Frame Regions”, dois rectângulos que quando tocados pelo castor deslocavam a imagem sobre este, mantendo-o sempre na região central do ecrã, o jogo tornou-se mais interessante. O problema de o urso ficar por vezes fora da área visual foi contornado com a introdução de um indicador (no protótipo inicial, uma mão, substituída posteriormente por uma seta) da posição relativa do urso. A distância foi indicada por um valor numérico. O GDD foi alterado de forma a reflectir o novo desenho e evoluiu a partir da ideia.

O sucesso da ideia inicial demonstrou que o projecto tinha valor, pelo que o conceito podia ser aprimorado. Com o desenvolvimento do conceito e da história foi possível criar uma boa mecânica de jogo. Após o criação da mecânica, os níveis começaram a ser elaborados. Com uma equipa de uma só pessoa as ideias para uma parte do trabalho vão surgindo com o desenvolvimento de outras áreas.

Durante toda a fase de desenvolvimento, a mecânica do jogo e alguns pormenores relativos ao nível criado para o protótipo final são aprimorados, e as alterações reflectidas no *Game Design Document* e no *Level Design Document*.

A criação de arte conceptual é o primeiro passo para o desenvolvimento da identidade visual do jogo. Com as duas fases acima descritas adiantadas, os elementos gráficos que vão constituir o jogo podem começar a ser criados e integrados em protótipos que começarão a transmitir a ideia de como resultará o jogo final. A resolução de cada objecto visual tem influência na jogabilidade final, o que faz com que a criação de *sprites* e a prototipagem estejam fortemente ligadas. São afinadas as velocidades relativas dos personagens, as trajectórias pré-definidas, as forças dos saltos e da gravidade. A animação tem pouca influência na mecânica do jogo mas tendo em conta o tempo disponível para a conclusão do projecto, é feita paralelamente com outras tarefas.

A música e os efeitos sonoros não estão tão fortemente interligados com as outras tarefas, pelo que ocupam a fase final do projecto. Nenhum destes elementos vai alterar a mecânica nem o desenvolvimento do jogo, mas sim contribuir para uma experiência mais agradável e imersiva.

Durante todo o processo de criação de protótipos, estes são testados e corrigidos até ser conseguida a experiência desejada. Os testes começam na fase inicial do projecto, juntamente com o primeiro protótipo, mas no final, testes mais exigentes são executados. Quando a experiência se revela satisfatória e não são encontrados erros internamente, o jogo (neste caso, o protótipo final) está pronto para ser distribuído a testadores externos, antes de entrar no mercado. Cada erro reportado é corrigido. A fase de testes e correcções, por ocupar um período variável dificilmente previsto, poderá ultrapassar o âmbito desta dissertação.

5.3 Ferramentas utilizadas

Várias ferramentas foram consideradas para o desenvolvimento deste projecto. Na maior parte dos casos foram usadas ferramentas comerciais devido à familiaridade do processo de trabalho e do interface. Em alguns casos a escolha foi também ditada pela possibilidade de criar e testar várias situações numa mesma ferramenta. No entanto, numa situação em que o acesso a estes programas comerciais se torna difícil ou caro, existe software *OpenSource* que ajuda no desenvolvimento e desempenha o seu papel tão bem como o aqui utilizado.

De todas as ferramentas possíveis, o papel e o lápis são sem dúvida as mais importantes. Estes dois utensílios acompanharam o desenvolvimento desde o seu início até ao seu término.

Para a criação do conceito, história, do *Game Design Document* e *Level Design Document*, qualquer processador de texto, ou até um simples editor são suficientes. No presente caso, o Microsoft Word foi a ferramenta de eleição por ser um programa fácil e poderoso, conhecido e de fácil acesso a estudantes. Noutra situação, o OpenOffice, StarOffice ou AbiWord poderiam ser usados.

Para a mecânica e desenho dos níveis, lápis e papel revelam-se como a ferramenta ideal. Também os elementos gráficos que fazem parte do jogo começaram por um esboço em papel que foi posteriormente digitalizado e vectorizado no computador. Alternativamente poderia ter sido usada uma mesa digitalizadora.

Os esboços a lápis passaram ao Adobe Flash para vectorização. Na maior parte dos casos este passo seria executado em software como o Adobe Illustrator, o Freehand ou, no caso do software gratuito, o Inkscape ou similar. A escolha do Adobe Flash deveu-se uma vez mais à familiaridade com o programa, mas especialmente ao conceito multifacetado da aplicação. Com a utilização deste programa, não só as imagens podem ser vectorizadas e coloridas, como é possível animar de forma quase automática os elementos não estáticos e ter uma ideia do resultado final. A opção de “*onion skin*” da aplicação facilitou o trabalho. O Flash pode ainda ser utilizado como ferramenta de prototipagem, utilizando *actionscript* para a interacção e acção não linear.

A maior parte dos elementos, especialmente os personagens vivos, partiram neste caso de um desenho feito a lápis sobre papel. Outros elementos, como camas e casas, tiveram origem em

modelos 3D importados para o Google SketchUp ou criados de raiz no software Cinema4D. Sobre estes modelos devidamente posicionados, os contornos e preenchimentos foram adicionados uma vez mais no Flash.

Com as várias imagens simples ou partes de animações criadas, o Photoshop foi utilizado para correcção e adição de detalhe. Os tamanhos e resoluções foram ajustados de forma a possibilitar a importação para o motor de jogo.

Seguindo o conceito de criação de um jogo sem nenhuma ou com o mínimo de programação, a opção da linguagem de programação e motor de jogo recaiu sobre o Game-Editor. Com um pagamento único inferior a \$100, este programa oferece uma licença vitalícia que permite a criação e venda de jogos, sem pagamentos extra. Utiliza uma linguagem com uma sintaxe semelhante à do C/C++ para a criação de scripts, e permite o desenvolvimento de código global em *Ansi C*. Existem outras alternativas, como o Game Maker, mas não permitem a exportação para as consolas abertas e telemóveis que constituíam o principal alvo deste projecto.

Os efeitos sonoros foram captados com um microfone de fontes reais. O tratamento foi feito com o programa Audacity, gratuito e semelhante à ferramenta profissional Audition, da Adobe.

A música foi criada novamente com as ferramentas essenciais, lápis e papel, sequenciada na aplicação Cakewalk Sonar, e reproduzida por meio do *SoftSynth* (software sintetizador) TTS-1, um *plugin* que obedece à norma *General MIDI*. Esta norma apresenta sons de instrumentos reais e artificiais cobrindo um leque bastante alargado e representando praticamente todas as famílias de instrumentos. Esta aproximação permite a utilização de sons com muito boa qualidade e simplifica a transformação da informação MIDI num ficheiro de áudio sem abandonar o domínio digital, mantendo portanto o máximo de qualidade. Embora o Game-Editor disponibilize a funcionalidade de reprodução de ficheiros MIDI, o baixo poder de processamento e a pouca memória das plataformas alvo deste projecto impossibilitam a geração de som com base em informação MIDI em tempo real. Ainda que tal fosse possível sem atrasar o jogo, uma grande vantagem desta aproximação é a homogeneidade do resultado. Um ficheiro de áudio PCM (*Pulse Code Modulation*) contém a informação do som sem qualquer perda e os sons resultantes não variam com a qualidade das formas de onda de cada equipamento. Como ponto negativo, a utilização de ficheiros de áudio em vez de informação MIDI incrementa em muito o tamanho do ficheiro do jogo. O espaço de armazenamento é no entanto bastante mais barato do que o poder de processamento, pelo que a aproximação representa na generalidade um balanço positivo. A utilização de um teclado MIDI, neste caso um Fatar SL-880, simplificou a introdução das notas no sequenciador através de um interface MIDI de baixo custo, o UM-1G da Roland.

5.4 Especificações para futuro

O projecto descrito foi focado na utilização em consolas, especialmente a GP2X e a GP2X Wiz. Para aproveitar a possibilidade de exportação para diferentes plataformas do Game-Editor, a utilização do produto final em terminais com Windows Mobile foi também estudada. Ao contrário das consolas, que apresentam características idênticas para cada modelo, estes equipamentos variam em termos de processador, memória, formato, resolução e controlos. A resolução do ecrã e

os controlos podem revelar-se insuficientes ou inapropriados para jogar o jogo conforme é descrito neste documento.

Em termos de resolução, no momento do desenvolvimento do projecto, existem três tipos característicos: 240x320, 480x640 e 480x800. O primeiro não apresenta qualquer problema, sendo possível ao Game-Editor exportar exactamente o mesmo projecto para ecrãs verticais, rodando a imagem 90°. Estes são os terminais Windows Mobile que deverão executar o projecto sem qualquer alteração com excepção da opção de exportação. Relativamente à resolução VGA, o Windows Mobile adapta de forma automática a resolução QVGA aos ecrãs de 480x640 pixéis. A grande vantagem de alterar o jogo para esta plataforma alvo seria o aproveitamento dos recursos oferecidos pelos equipamentos com esta resolução. Todos os objectos visuais do jogo foram criados com resoluções bastante superiores aquela que utilizam no jogo. A adaptação a ecrãs VGA passaria por um novo dimensionamento e correcção de pormenores para que estes mantivessem o máximo possível de detalhe. Os níveis teriam também que ser recriados de forma a reposicionar os elementos para o quádruplo da sua resolução. Em termos de resoluções WVGA, o mesmo processo seria necessário. A “view” (janela que apresenta o conteúdo do ecrã) seria maior mas isso não teria implicações negativas na jogabilidade. O facto de o Game-Editor não apresentar a resolução de 480x800 como opção de exportação para PocketPC não deveria ser grande obstáculo, visto que a exportação para HandheldPC deverá resultar num executável que corre em PocketPCs. Alternativamente, a resolução do ecrã de 480x800 é suficiente para apresentar um jogo em 480x640 com barras negras laterais.

Os controlos de jogo podem representar um problema maior do que as questões mencionadas anteriormente. Os métodos de entrada variam com os equipamentos e nem todos apresentam o número de teclas necessário à execução do jogo conforme previsto. Mesmo a existência de teclas direccionais não pode ser garantida, e equipamentos como o HTC Touch Diamond ou Touch HD não dispõem das referidas teclas. Outros, como o Samsung SGH-i780, apresentam um pequeno quadrado sensível ao toque que pode funcionar como controlador para rato ou como entrada direccional, conforme as definições do telemóvel.

A solução mais rápida passaria por controlos virtuais no ecrã, como os existentes em alguns emuladores ou em jogos como “Tomb Raider” para PocketPC. Formas geométricas são sobrepostas ao jogo apresentando apenas os seus contornos ou um certo grau de transparência. Quando o jogador pressiona o ecrã sobre o ponto onde se encontra um destes controlos virtuais, o jogo reconhece o toque como se fosse uma pressão de uma tecla, respondendo em conformidade.



Fig. 68 – Tomb Raider – Botões virtuais em Windows Mobile

Nas consolas, a disposição dos botões é otimizada para o jogo. Nos telemóveis, esta disposição é otimizada para as funções mais comuns, como atender o telefone ou abrir o programa de correio electrónico. A possibilidade de redefinir as teclas no telemóvel seria uma mais-valia e permitiria também utilizar botões de volume, por exemplo, como controladores do jogo. Para melhorar ainda mais a experiência, os botões virtuais seriam também programáveis e só apareceriam no ecrã se fossem realmente necessários.

Uma alternativa à falta de botões ou má disposição destes para o jogo seria a alteração da interface de forma a permitir a interacção com o rato. Enquanto o número de botões é variável entre os modelos com Windows Mobile, a existência de um ecrã sensível ao toque, resistivo, é uma constante no Windows Mobile Professional. Um novo planeamento do método de interacção seria necessário, mas a nova solução garantiria a mesma experiência de jogo em qualquer terminal deste tipo. As consolas GP2X-F200 e GP2X Wiz beneficiariam também deste método de entrada de dados.

No jogo podem existir uma ou várias plataformas distribuídas sobre o eixo vertical do mesmo. O comprimento das plataformas ou do nível é limitado apenas pelas definições do *designer* de cada nível, podendo o fundo deslizar para a esquerda ou para a direita de forma a permitir a visualização de todo o nível. Não existe, no entanto, movimento do fundo sobre o eixo vertical. Quando o castor se aproxima das margens esquerda ou direita, o fundo desliza no sentido contrário, mas o seu movimento permanece inalterado quando o castor sobe ou desce de umas plataformas para outras. O motor de jogo poderia ser alterado de forma a possibilitar um movimento na vertical, permitindo níveis maiores em altura, com maiores graus de complexidade.

6. Implementação

Com o planeamento completo é necessário começar a desenvolver o produto final. A ideia geral do jogo e da sua mecânica serve como ponto de partida para especificar cada um dos níveis e demais elementos do jogo. Várias competências diferentes são envolvidas de forma a criar os diversos componentes que serão montados no jogo final.

6.1 Idealização

As funções do designer do jogo compreendem o desenvolvimento do conceito, da história, a criação da mecânica do jogo e o desenho dos níveis. Com as três primeiras tarefas praticamente completas na fase de planeamento, o desenho do nível utilizado no protótipo final sofreu ligeiras alterações após as primeiras experiências práticas. Depois de jogar o nível conforme tinha inicialmente sido planeado, tornou-se evidente que algumas situações deveriam ser revistas. Era bastante fácil terminar o nível, que no entanto apresentava elementos a mais. O morcego viria a desaparecer do nível inicial, e as posições dos elementos foram alteradas de forma a maximizar a componente de “puzzle” do jogo. Apesar da acção não ser completamente linear, permitindo por vezes completar um nível de formas diferentes, a ideia do jogo assenta no facto de que o jogador deve pensar na estratégia correcta.

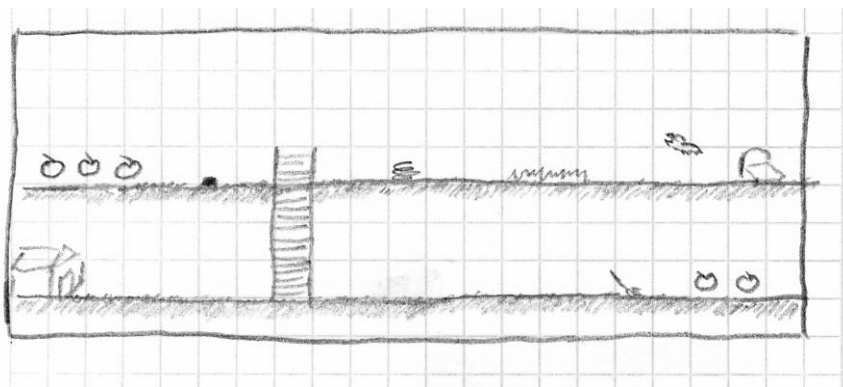


Fig. 69 – Desenho inicial do nível da Floresta

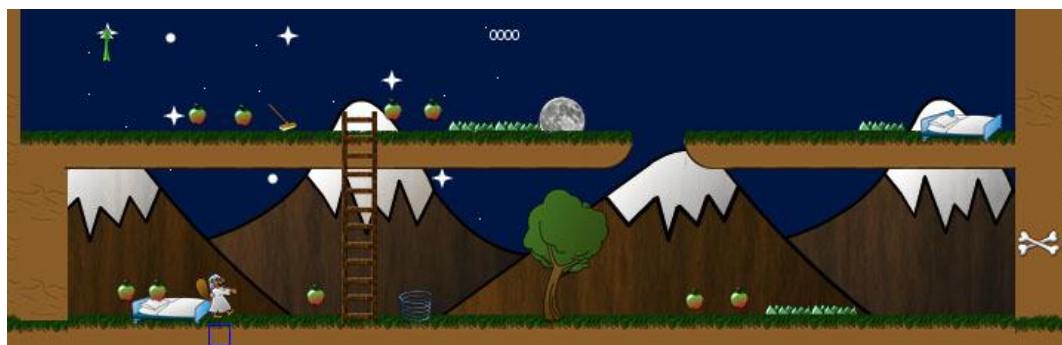


Fig. 70 – Resultado final do nível da Floresta

As maçãs foram redistribuídas por locais estratégicos, foi alterado o número e posicionamento dos vidros, a mola e a vassoura passaram a ocupar novas posições. Com a saída do morcego, a pedra deixou também de ser necessária. O resultado foi uma distribuição mais limpa dos elementos do jogo, tornando-o menos confuso, e simultaneamente uma experiência que exige alguma perícia e raciocínio.

Com a nova definição do nível, vários elementos da mecânica do jogo foram demonstrados, como as quedas de lugares elevados, o funcionamento da mola e da vassoura e o aumento temporário da velocidade trazido pelas maçãs.

Os limites esquerdo e direito receberam paredes que bloqueiam os heróis, tal como o chão, de forma a evitar comportamentos indesejados como a queda do urso ou do castor para fora da área de jogo.

Embora exista a possibilidade de haver várias maneiras de resolver o puzzle, nenhuma é óbvia à partida e é necessário o planeamento cuidadoso para a resolução do *puzzle*. Com o intuito de demonstrar correctamente os vários elementos da mecânica do jogo, o nível desenvolvido para o protótipo final apresenta alguma complexidade, não ocupando uma posição no início do jogo final mas sim no fim do primeiro mundo – a Floresta.

6.2 Arte

Para começar a definir a identidade visual do jogo, foram feitos alguns desenhos dos personagens. Uma vez mais, o lápis e a borracha revelaram-se ferramentas essenciais. Os dois personagens foram os primeiros a ser criados, e enquanto o resultado inicial foi satisfatório para o desenho de Godofredo, a imagem de Inácio ficou definitiva à terceira tentativa.



Fig. 71 – Arte Conceptual – Godofredo

Para desempenhar o papel de melhor amigo de Godofredo, vários animais foram considerados. Um panda, um guaxinim e um castor. O panda foi a primeira escolha por ser também um urso, mas o facto de o seu habitat natural ser diferente do da espécie de Godofredo deu origem a mais experiências. Embora o jogo se desenrole num ambiente de fantasia em que os ursos vivem em cabanas e usam gorros para dormir, um panda pareceu um pouco desenquadrado. O guaxinim necessitava de bastante detalhe para ser reconhecido como tal e acabou por ser substituído pelo castor. As manchas escuras em volta dos olhos e a cauda listada do primeiro esboço transmitiam também uma conotação negativa, assemelhando-se a um dos “Irmãos Metralha”, com a característica máscara e as listas semelhantes a um uniforme de prisão. Após seleccionado o castor, os tamanhos relativos dos dois heróis foram alterados para que não se apresentassem desproporcionados no ambiente de jogo.



Fig. 72 – Arte Conceptual – Ideia original para o adjuvante de Godofredo

Devido ao mundo de fantasia em que o jogo se enquadra, algumas liberdades foram permitidas e o realismo foi deixado para segundo plano. De forma a dar uma imagem alegre e jovial ao jogo, ambos os personagens foram vestidos. Aquilo que começou por ser um par de patas em cada um dos heróis deu lugar aos pés humanos de Godofredo e às sapatilhas desportivas de Inácio.

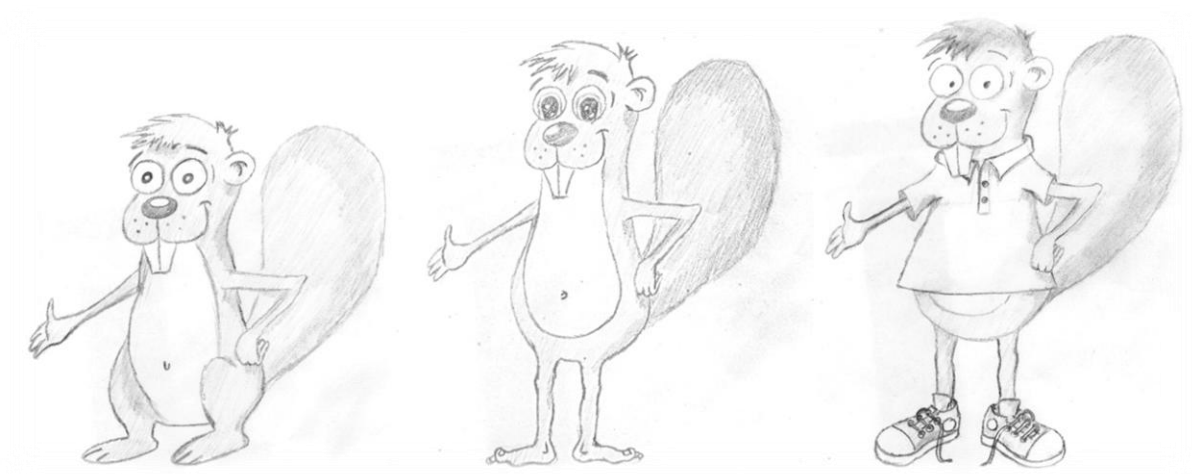


Fig. 73 – Arte Conceptual – Inácio

O habitat dos ursos e castores foi ligeiramente alterado de forma a misturar a ideia da realidade com a de personagens carismáticos com os quais o jogador poderá desenvolver uma certa proximidade e o máximo possível de sentido de protecção. Para atingir este objectivo, o urso vive numa cabana da floresta, o que se enquadra também com a túnica e o gorro de dorminhoco.



Fig. 74 – Arte Conceptual – Casa do Urso Godofredo

Foram também criadas imagens para aranhas, cobras, morcegos, ouriços e vários outros personagens que viriam ou não a integrar a aventura. Independentemente de aparecer no produto final, todos os elementos contribuíram para uma visão mais imersiva do mundo em que o jogo se passa. Alguns destes elementos acabaram por se transformar quase directamente em *sprites*, sem grandes edições, enquanto outros serviram apenas de modelo ou de inspiração para os gráficos finais do jogo.

A arte conceptual originou a identidade visual do jogo que foi aprimorada durante a fase da criação de *assets*. Os desenhos a lápis foram digitalizados e importados para o Adobe Flash. Nesta ferramenta foi traçado inicialmente o contorno de cada imagem, com linhas e curvas, passando depois para o preenchimento. Apesar da luz difusa que em situação real seria proveniente das estrelas e da lua, o preenchimento teve em consideração uma fonte de luz posicionada acima e à esquerda da acção, no lado do observador sobre o eixo perpendicular ao ecrã. Esta medida deu aos elementos do jogo uma ideia de profundidade tornando a visualização mais interessante, ainda que menos realista.

6.3 Programação

Pela possibilidade de criação de jogos sem código e a automatização de várias funções com ferramentas visuais, pelo limite de tempo disponível para este projecto, e ainda pela possibilidade de exportação de jogos para as consolas aqui visadas, o programa Game-Editor foi escolhido quase no início do projecto para implementar o jogo e a sua mecânica. O intuito do projecto é também a criação de um jogo com o mínimo possível de código, pelo que neste ponto a ferramenta se revelou mais uma vez uma boa escolha.

Com a utilização do Game-Editor, aquilo que na criação de outro jogo seria conseguido com recurso a código é, na maior parte das vezes, desenvolvido com reacções a eventos que a aplicação disponibiliza.

O Game-Editor apresenta uma grelha sobre a qual o jogo pode ser desenhado. Todos os objectos que fazem parte do jogo, sejam *sprites* animadas, fundos ou textos, têm o nome de “Actors” e funcionam como objectos numa linguagem orientada a objectos. Apresentam métodos, propriedades e eventos para controlo e interacção com todos os outros actores. Para cada actor podem também ser utilizadas várias instâncias que obedecem às mesmas regras, simplificando a duplicação de actores do mesmo tipo, como nos casos das maçãs, dos vidros ou das camas. Além dos comportamentos estáticos relativos a cada actor, ou objecto, as instâncias de um mesmo actor podem utilizar também comportamentos específicos. Esta funcionalidade revelou-se bastante útil permitindo, como exemplo, a existência de duas camas em cada nível (a cama de onde o urso sai e a cama onde o urso deve chegar para terminar o nível), mas terminando o nível apenas quando o urso atinge a cama de destino.

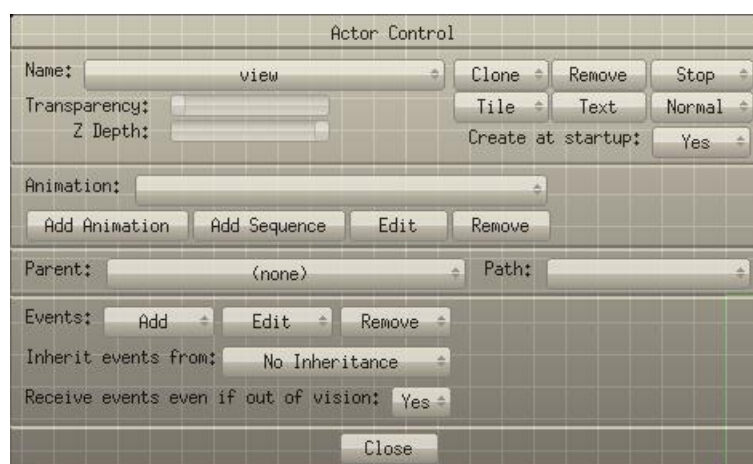


Fig. 75 – Game-Editor – Caixa de diálogo para controlo dos actores

A aproximação do Game-Editor ao desenvolvimento de jogos simplifica o processo, à custa de alguma organização. Às vezes é difícil saber onde está a lógica que provoca certas acções, pelo que um bom planeamento e documentação são extremamente importantes.

Para contribuir para a imersão e dotar o jogo de uma sensação de tridimensionalidade, três artefactos foram utilizados: o preenchimento com base numa fonte de luz imaginária; a ligeira inclinação de alguns elementos sobre o eixo vertical criada na fase do desenho dos elementos gráficos; e o efeito de paralaxe obtido pela utilização de quatro planos de fundo com velocidades diferentes. Como este mecanismo é específico do jogo em questão e não está previsto pelo Game-Editor, a introdução de código foi necessária para atingir o efeito pretendido. Dos quatro níveis de movimento utilizados, o fundo mais distante (contendo estrelas e objectos distantes no céu) permanece estático, pelo que não necessita de qualquer alteração. Os outros três níveis contêm respectivamente os elementos mais próximos no céu (lua, nuvens), as montanhas ao fundo e os objectos em primeiro plano, onde se passa a acção. O código a utilizar é extremamente simples, sendo apenas necessária a actualização das posições dos respectivos actores com diferentes valores correspondentes à velocidade de cada um. A propriedade “x” contém a posição horizontal do actor, e cada actor é referenciado pelo seu nome. O código utiliza “dot notation” e é igual ao utilizado em linguagens orientadas a objectos:

```
moon.x-=0.2;
```

```
mountains.x-=0.5;
```

```
ground.x-=1;
```

Estas alterações são introduzidas na gestão de um evento que ocorre trinta vezes por segundo e cuja taxa de refrescamento é definida nas propriedades do jogo. O efeito é um movimento de um pixel a cada 1/30 segundos para o primeiro plano, meio pixel para as montanhas e um quinto de pixel para a lua e/ou nuvens.

Os diferentes níveis de movimento contêm vários elementos gráficos diferentes e são montados no editor do Game-Editor. A aproximação por “tiles” permite poupar muita memória evitando o registo de todos os pontos de padrões idênticos. Como resultado, o chão sobre o qual se movem os actores apresenta erva à superfície, alguns ossos e irregularidades a um nível mais baixo, mas todo o chão é tratado como um só elemento e todos os “tiles” fazem parte de um mesmo actor.

Para a funcionalidade da seta que aponta a direcção de Godofredo, e porque o Game-Editor não permite a rotação de gráficos no ecrã de forma nativa, a técnica utilizada assentou na criação de uma animação com a seta a apontar em diferentes direcções. O código que se encarrega da actualização da seta traduz o ângulo a representar numa posição da animação que contém a seta.

```
double angle = 360-direction(xscreen, yscreen, sw.xscreen, sw.yscreen);
```

```
animpos=(angle/360)*(nframes-1);
```



Fig. 76 – Animação da seta que aponta na direcção de Godofredo

Os inimigos móveis seguem caminhos pré-definidos no Game-Editor. Em certas situações, apesar do caminho definido para um actor ser constante, certas alterações ocorrem no decorrer desse caminho. O Game-Editor disponibiliza o acesso a eventos como o final de uma trajectória, e esses eventos podem ser utilizados para fazer as alterações necessárias. Tomando como exemplo o caso do morcego, este segue um caminho em forma de número “8” rodado 90°. A posição do morcego aponta para a esquerda na primeira metade da trajectória, mas deve inverter-se para completar a segunda metade. Neste tipo de situações, foram criadas duas trajectórias em vez de uma. Ao iniciar o trajecto, o morcego aparece virado na direcção em que voa. Quando é atingido o fim dessa trajectória, o Game-Editor disponibiliza o evento “*Path Finish*”, no qual é alterado o caminho para a outra metade da figura “8”, e também a animação utilizada. O morcego acabou por ser retirado do nível do protótipo para aumentar a jogabilidade do mesmo, mas este comportamento está patente também em situações como a perda de uma vida. Quando o urso toca no chão com uma velocidade vertical superior a um número pré-definido (devido à aceleração da gravidade) ou quando toca nos vidros, é reproduzido um som, é alterada a sua animação e a sua trajectória. No final desta trajectória, em formato de “U” invertido, uma vida é decrementada e a janela do jogo é reposicionada no seu ponto inicial.

Cada actor mantém e actualiza as suas propriedades independentemente da maneira como está a ser apresentado no ecrã. A representação pode ser alterada entre as várias animações atribuídas a cada actor. Esta característica é utilizada também em situações como a perda de uma vida, em que a animação lateral e o controlo pelo teclado é substituída por uma visão frontal do herói que originou a perda da vida, a abanar os braços. O seu movimento também passa a um caminho pré-definido, tudo isto na continuação de um evento de colisão. No fim da trajectória pré-definida é despoletado um evento que é utilizado para decrementar o número de vidas ou testar o fim de jogo, caso o jogador não disponha de mais vidas.

Para evitar atrasos por excesso de elementos em jogo e sobrecargas de memória (factos importantes em terminais móveis com baixo poder de processamento e quantidade de memória), o Game-Editor disponibiliza uma ferramenta que dá pelo nome de “*Activation Region*”. Um rectângulo é desenhado em torno de determinadas áreas do ecrã e todos os actores que não intersectarem o rectângulo que também intersecta a *view* permanecem inactivos. Estas regiões revelam-se extremamente úteis sempre que é necessário reiniciar um nível. Teoricamente é suficiente mudar a janela de jogo para um ponto que não intersecte a região do nível e voltar a repor esta janela no seu início, e todos os elementos do nível que não tiverem sido explicitamente destruídos voltam a ser carregados na memória, nas suas posições iniciais.

Sendo também uma ferramenta desenvolvida por um programador *indie*, o Game-Editor apresenta alguns “*bugs*”. Devido à execução simultânea de vários processos, por vezes alguns eventos não eram executados antes da reposição da janela. Para evitar problemas no jogo, foi criado um “*timer*”, ou um contador. Desta forma, quando termina o caminho em forma de “U” invertido, a janela de jogo é movida para fora da acção e o contador é iniciado. A janela é movida de volta para o início do nível apenas quando o contador termina a sua contagem.

Uma funcionalidade extremamente útil do Game-Editor permite definir o “*parent*” ou “*pai*” de cada actor. Ao alterar o “*parent*” da vassoura, por exemplo, para o castor Inácio, a posição da vassoura é actualizada automaticamente sempre que é actualizada a posição do castor. Desta forma

é desnecessário quase qualquer código relativo à vassoura. Quando ocorre uma colisão do castor com a vassoura, o “parent” desta é alterado para o castor. Quando ocorre a colisão da vassoura com os vidros, é destruído o actor que representa os vidros. Com este tipo de mecanismos, uma grande parte da programação é substituída por reacções a eventos, programadas de uma forma visual com apenas duas ou três pressões no botão do rato. No presente caso, a programação de alguma lógica é ainda assim necessária, de forma a alterar a posição e a visualização da vassoura em situações como a mudança de direcção do castor. É assim possível que a vassoura esteja sempre localizada na frente de Inácio e orientada na sua direcção. A quantidade de código necessária para esta situação é, ainda assim, bastante inferior à quantidade de código necessária para realizar as mesmas acções em C++.

6.4 Sprites e animação

As imagens que são utilizadas no jogo têm a sua origem em diferentes conteúdos. Na maior parte das situações, desenhos a lápis baseados na arte conceptual desenvolvida serviram como base a uma vectorização que foi posteriormente colorida. Alguns dos desenhos criados na fase de desenvolvimento da arte conceptual apresentavam já as características necessárias para serem transformados em *sprites*, como o caso do desenho do urso Godofredo. O detalhe apresentado por uma figura de pequenas dimensões é bastante inferior ao que apresentam as imagens originais, devido à perda de resolução resultante. O detalhe é também menos importante porque a atenção do jogador vai estar focada no jogo e não na ilustração.

Tendo em conta que as várias fases da criação do jogo foram desenvolvidas pela mesma pessoa, a função do animador fundiu-se com a do ilustrador. A opção da aplicação Flash da Adobe permitiu animar os elementos na mesma ferramenta em que foram criados, tornando praticamente ambas as funções numa só.



Fig. 77 – Godofredo vectorizado e preenchido a partir da arte conceptual

O fundo verde foi utilizado na criação e especialmente no preenchimento das *sprites* para ser possível distinguir facilmente entre o fundo e outros pormenores que apresentam a mesma cor que este, como o brilho no nariz do urso em fundo branco. A cor utilizada foi escolhida por não existirem elementos no jogo de cor semelhante e por revelar bastante contraste com qualquer tonalidade utilizada.

As animações foram também desenhadas em papel utilizando a técnica de “*Onion Skin*”, ou casca de cebola. Novos desenhos foram traçados em folhas translúcidas dispostas sobre o desenho anterior, sobre um vidro com iluminação posterior. Com todas as imagens constituintes de cada animação importadas para o Flash, a funcionalidade de “*Onion Skin*” do próprio Flash foi utilizada para dispor correctamente cada imagem sobre a anterior (Ferreira and Azevedo 2001). Partes comuns foram copiadas e as alterações foram desenhadas novamente. Após a vectorização de todas as imagens, e após o teste da animação de forma a minimizar saltos artificiais, estas foram dispostas na mesma *frame* da *timeline* e coloridas. Com a utilização de guias, as várias imagens foram dispostas com precisão para que os seus centros de massa ficassem equidistantes uns dos outros. O documento foi também criado com dimensões múltiplas das de cada imagem. Esta aproximação é necessária porque o Game-Editor considera um mapa de *bits*, contendo todas as *frames* da animação, como se fosse uma sequência de imagens de igual tamanho. Para animações mais simples, como o castor a subir escadas, duas imagens foram criadas, enquanto para obter um movimento fluído, elementos como o morcego utilizaram seis imagens em diferentes posições.

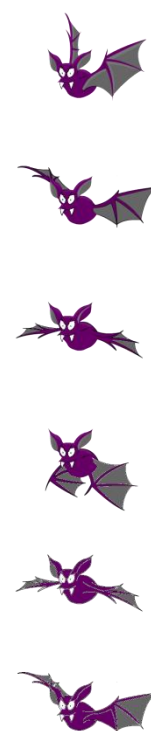


Fig. 78 – Animação final do morcego

Elementos assimétricos que no jogo se movimentam em dois sentidos foram coloridos independentemente. Os contornos mantiveram-se, para que a animação do castor, do urso, do morcego ou da vassoura apresentem as mesmas linhas rodadas sobre um eixo vertical para dar origem à mesma animação no sentido contrário. O preenchimento foi, no entanto, alterado para manter a fonte de luz no mesmo ponto.

Alguns elementos, como a cama ou a casa, foram criados com base em modelos tridimensionais existentes. Pela facilidade de utilização e pesquisa de objectos, o Google SketchUp foi utilizado para o posicionamento dos modelos. O facto de a aplicação não ter qualquer custo também faz dela uma boa escolha. Estes modelos serviram apenas como base e foram também vectorizados e preenchidos no Flash, tal como todos os elementos. Com esta reutilização da mesma ferramenta foi possível fazer com que todos os elementos partilhassem uma linha comum, facilitando a sua integração no nível. Alguns elementos gráficos podem ser criados mais rapidamente num programa de modelação 3D, especialmente objectos fortemente relacionados com primitivas ou NURBS (*Non-Uniform Rational B-Spline*). Utilizando como exemplo a mola que projecta o urso na vertical, foi utilizada uma espiral de cinco voltas, ou 1800° . O que seria relativamente complicado de executar manualmente com o lápis sobre papel foi rapidamente modelado fazendo um pequeno círculo seguir um percurso helicoidal com a utilização da ferramenta *Sweep NURBS* do programa Cinema4D. Uma vez mais, o resultado da renderização do programa foi exportado para o Flash onde os contornos foram desenhados e o preenchimento adicionado.



Fig. 79 – Mola modelada no programa Cinema4D

O Photoshop foi utilizado para confirmar o alinhamento das imagens, adicionar ou corrigir detalhes, remover o fundo e finalmente exportar como PNG (*Portable Network Graphics*). O formato PNG não só permite uma compressão sem perdas como também mantém o fundo transparente após a exportação, o que possibilita a sobreposição de elementos no Game-Editor sem artefactos indesejados.

6.5 Som

A introdução do som contribui para uma maior imersão e para uma maior sensação de interação com o jogo. Quase qualquer acção, desde o folhear de um livro à abertura de uma porta, produz um som. Ao reproduzir estes fenómenos num jogo, é possível aumentar o realismo e proporcionar ao jogador uma melhor sensação de reacção por parte do jogo.

A música ajuda a desenvolver o ambiente do jogo, e pequenas melodias produzem efeitos que se adequam à acção enquanto esta decorre. Acordes ou harpejos maiores transmitem uma certa alegria, acordes ou harpejos menores representam tristeza. Harpejos ou intervalos ascendentes indicam algo positivo, como atingir um objectivo, enquanto os descendentes representam situações negativas. Outras características musicais podem introduzir tensão, relaxamento, calma ou até sentimentos de medo ou receio.

Apesar de não ser essencial para os testes da mecânica ou do código que gera o jogo, a música e o som têm um forte impacto na jogabilidade e na imersão proporcionadas.

6.5.1 Música

Seguindo as especificações do *Game Design Document*, a música foi criada tendo sempre em mente a simplicidade. Numa experiência como um jogo, em que vários sentidos são utilizados numa só actividade, é possível haver excesso de informação que pode levar a confusão e falta de concentração na actividade principal. A música é uma parte extremamente importante em qualquer experiência multimédia, mas no jogo a ideia principal continua a ser fazer com que o urso, partindo do ponto A, alcance o ponto B. Melodias, harmonias e padrões rítmicos simples apresentam também melhor probabilidade de ser bem recebidos por grande quantidade da população.

Uma das especificações do *Game Design Document* dita que a música deve ser alusiva ao mundo do nível em que é reproduzida. No caso do protótipo apresentado, e tendo em conta que existem vários tipos de floresta, foi considerado o *habitat* natural do castor no qual se podem também encontrar ursos da espécie de Godofredo. Partindo da ideia transmitida regularmente pela televisão, uma das principais fontes de informação para grande parte da população do globo, instrumentos típicos do *habitat* do *Castor Canadensis* foram utilizados: o banjo e a harmónica. A harmónica reproduz a melodia principal dobrada pelo som da *Music Box*. Este instrumento foi seleccionado não pela sua relevância geográfica mas sim pelo carácter percussivo que ajuda a marcar as notas. A *Music Box* é um instrumento musical pré-programado que reproduz a música por meio de pequenas elevações num cilindro que activam palhetas de metal. Este instrumento é muitas vezes programado com canções de embalar, facto que contribui para o ambiente em que o urso caminha enquanto dorme.

O banjo reproduz a melodia principal durante a secção B do tema em uníssono com a harmónica. Durante toda a música este instrumento executa também um padrão característico do estilo *Ragtime*, nascido em cidades como St. Louis e New Orleans, o *Secondary Rag*. Este padrão resulta da utilização de um motivo de três notas sobre um compasso quaternário (Boffi and Morão 1999).



Fig. 80 – Melodia e contra melodia, Secondary Rag e contrabaixo no tema da primeira parte da secção A da floresta.

A contra melodia foi introduzida de forma a preencher os espaços deixados pela melodia principal. Além dos motivos estéticos, a introdução desta segunda melodia permite reduzir silêncios. Apesar da inegável necessidade do silêncio na música, esta aproximação ajuda a transmitir a necessidade de continuar o jogo e a incutir o ritmo em que se pretende envolver o jogador.

A tonalidade de Dó Maior foi escolhida pela sua familiaridade e foi mantida durante a quase totalidade do tema. A única excepção encontra-se no final da secção B, onde a introdução de um Fá # no acorde de D7 sugere temporariamente uma modulação para Sol Maior. Este acorde serve apenas como dominante secundário e acrescenta alguma variedade a um tema que apresenta uma duração bastante curta para não ocupar muito espaço no suporte digital.

6.5.2 Efeitos Sonoros

Os efeitos sonoros para o jogo foram capturados com um microfone de condensador estereofónico e posteriormente processados com a ferramenta Audacity.

Um criador de jogos independente nem sempre tem acesso ao material profissional utilizado pelas grandes empresas e é por vezes necessário recorrer a métodos mais tradicionais para a criação de certos conteúdos, sendo o som um exemplo típico.

Dada a impossibilidade de captação do som em câmara anecóica, foram tomadas providências para reduzir ao mínimo as reflexões e reverberações e os ruídos externos. As paredes e vidros foram, dentro do possível, tapadas com cobertores e cortinas. Portas e janelas foram fechadas de forma a isolar ao máximo os ruídos provenientes do exterior, os aparelhos eléctricos foram desligados à excepção do dispositivo de captura de som, e o cabo do microfone foi mantido com um comprimento mínimo.

Os ficheiros de áudio foram gravados num cartão SD por intermédio de uma câmara de vídeo digital Sanyo Xacti HD1000. Na falta de um dispositivo dedicado de gravação de áudio, esta

câmara oferece um modo de captura de dois canais a uma frequência de 48KHz e com 16 *bits* de quantização, no formato AAC (*Advanced Audio Coding*). Por ser bastante pequena foi possível utilizar um cabo de microfone com cerca de dez centímetros de comprimento, e por não utilizar partes móveis no modo de gravação de áudio foi possível eliminar ruídos de *zoom* ou de ventoinhas. O som resultante continha ainda assim uma quantidade audível de ruído, ruído este que foi posteriormente reduzido com utilização do programa de tratamento de áudio.

Os efeitos sonoros foram, quando possível, provenientes de objectos que produzem o som a utilizar no jogo. O som da vassoura foi capturado a partir de uma vassoura real a varrer o chão, e o som da maçã é também o som real de dentes a trincarem o fruto. O som da mola foi conseguido pela utilização de um “berimbau de boca”, instrumento que gera um som metálico e elástico. Aproveitando uma das vantagens da gravação digital, várias amostras foram gravadas do mesmo som e a melhor foi escolhida e posteriormente tratada.

O programa Audacity foi utilizado para cortar o som nos pontos correctos depois de ter escolhido a melhor amostra. Apesar de todo o cuidado na captação, as amostras de som apresentavam algum ruído que foi retirado com a ferramenta de redução de ruído do Audition. Para este efeito foram gravados alguns segundos de silêncio antes de cada amostra de som, silêncio este que serviu de amostra de ruído. De forma a evitar um resultado artificial, a redução de ruído foi aplicada utilizando os parâmetros mais baixos que produziam um resultado satisfatório.

Após o acerto dos pontos inicial e final de cada amostra de som e a redução de ruído, todos os sons foram normalizados, mantendo assim um nível de saída constante para os diferentes efeitos. Nos casos das vozes do urso ou do castor, foram utilizadas gravações de voz real, tentando emitir um som agudo para a captação. Estes sons foram posteriormente processados com uma ferramenta de alteração da altura do som, dando origem a vozes fictícias concordantes com as imagens dos personagens.

7. Avaliação

Após a conclusão do jogo é necessário um período de testes que permite confirmar a funcionalidade do mesmo. No presente caso, o protótipo foi testado pelo autor mas, como foi anteriormente referido, uma equipa de testes externa pode confirmar com maior neutralidade aquilo que resultou e apontar eventuais falhas. Paralelamente é analisada a aceitação do protótipo por esta mesma equipa, heterogénea e representativa do público-alvo do jogo.

Os testes realizados sobre o protótipo foram pensados de forma a responder a dois tipos distintos de questão. Por um lado, um inquérito serviu para recolher respostas relativas à aceitação e opinião geral dos inquiridos sobre o jogo desenvolvido. Por outro lado, respostas directas de um grupo de testes permitiram detectar “bugs” de forma a possibilitar a correcção de situações anteriormente desconhecidas.

7.1 Aceitação

Com dois objectivos principais, este trabalho consistiu na análise da possibilidade de criação de um jogo por uma equipa de uma ou poucas pessoas tendo simultaneamente em atenção a originalidade deste mesmo jogo. O desenvolvimento do trabalho assentou sobre um estudo de jogos disponíveis e numa experiência pessoal com inúmeros jogos de diferentes estilos ao longo de vários anos. É de salientar contudo que a ideia de originalidade pode variar entre diferentes pessoas.

Embora apenas pouco mais de 13% das respostas ao inquérito considerem o jogo completamente original, a maioria das pessoas classificou-o como contendo “ideias novas suficientes para captar a atenção”. Menos de um quarto das respostas indica que o jogo contém alguns elementos originais sem oferecer nada de novo e uma única resposta aponta-o como “igual a muitos outros”. No geral, a ideia de originalidade conseguiu transparecer e apesar de não representar um novo “Tetris”, os resultados apresentaram-se bastante satisfatórios.

Ao contrário dos jogos comerciais, o projecto foi desenvolvido por uma só pessoa, com vários interesses distintos dentro do universo dos media mas sem pretensões de se equiparar em termos de capacidade de criação de conteúdos a um especialista de cada uma das áreas. Foi, portanto, uma recompensa gratificante a aceitação e apreciação dos conteúdos visuais. A maior parte das respostas classificaram os elementos gráficos como bons para o tipo de jogo em questão e mais de um quarto apontaram-nos mesmo como “muito bem concebidos”. As restantes pessoas consideraram o aspecto gráfico como básico mas interessante e ninguém o considerou feio e mal ilustrado. Paralelamente à qualidade gráfica foi analisada a relação entre o tipo de jogo e a imagem que este apresenta. Neste ponto as respostas apresentaram-se ainda mais positivas, tendo em conta que menos de um terço considerou que haveria melhores escolhas – mas ainda assim reportando a ligação entre os gráficos e o jogo como bem conseguida. As restantes respostas consideraram a combinação exactamente adequada.

Ainda dentro do aspecto gráfico mas em termos de movimento, ninguém apontou a animação como básica e rudimentar. As respostas dividiram-se entre as quatro opções mais favoráveis com a grande maioria a considerar os movimentos fluidos e naturais.

Os efeitos sonoros apresentaram um maior problema na sua criação por falta de espaço e material dedicado, duas questões com que uma equipa profissional normalmente não se depara. Apesar do estúdio improvisado num quarto com cobertores e equipamento amador de captação, sensivelmente três quartos dos inquiridos atribuíram nota máxima a este elemento. Os efeitos sonoros receberam nota positiva em termos de captação e processamento, assim como na sua relação com o tipo de jogo. Um quinto das pessoas detectou alguns problemas, relacionados provavelmente com os níveis de redução do ruído provenientes do material de captação. A relação dos efeitos com o jogo foi, no entanto, considerada satisfatória. Apenas 5% consideraram mal conseguida a ligação entre os sons e o jogo, o que mais uma vez suporta a ideia de criação de um jogo completo por uma só pessoa.

A segunda metade do inquérito revelou-se a mais motivadora por se relacionar com factores de mecânica e de possibilidades comerciais, talvez as parte mais importantes em qualquer jogo. A jogabilidade foi considerada como “cativante” por uma maioria absoluta e apenas duas pessoas responderam que jogariam o jogo apenas para experimentar. Os restantes consideraram que o jogo precisa de muito trabalho embora contenha ideias interessantes. O projecto não visava a produção de um jogo completo mas sim das suas especificações e apenas um nível foi disponibilizado, pelo que uma vez mais os resultados se revelaram extremamente animadores.

De forma a demonstrar vários elementos do jogo foi estipulado à partida que o nível apresentado seria uma das etapas finais de uma área do jogo, prescindindo de tutoriais e da habituação progressiva à mecânica. Por este motivo resultou numa certa surpresa a resposta de quase 18% dos inquiridos que consideraram o *puzzle* fácil de completar. A maioria confirmou o resultado esperado de que o nível é bastante difícil, porém adequado ao final de um dos mundos. Duas pessoas consideraram o nível praticamente impossível, enquanto os restantes confirmaram a dificuldade de resolução do problema mas apontaram a necessidade de perícia de movimentos como baixa.

As respostas às duas perguntas mais indicativas da aceitação do projecto indicaram-no quase unanimemente como normal dentro do tipo de plataforma a que se destina e, quase metade dos inquiridos consideraram o jogo equiparável às produções de estúdios profissionais. Apenas duas respostas consideraram o jogo como amador. Em termos de possibilidade de distribuição comercial do jogo, apenas 11.3% indicaram que este deveria ser distribuído gratuitamente enquanto todos os demais estariam supostamente dispostos a pagar. Apontando uma vez mais o facto de que este jogo nunca foi encarado numa perspectiva comercial, a ideia de um tão grande leque de potenciais interessados na compra do jogo aponta para o sucesso da ideia.

Finalmente, em termos de erros de desenvolvimento, a maioria considerou a experiência livre de “bugs”. O facto de este grupo não ter detectado erros não significa que eles não existam, mas sim que não se revelaram na execução do nível do jogo. Como protótipo, é normal a existência de algumas falhas que foram corrigidas, dentro do possível, após a análise do inquérito e a troca de ideias com o grupo de testes.

Por muito animadores que se apresentem os resultados do inquérito, estes só serão considerados válidos quando comparados com os do inquérito inicial que analisava a viabilidade do projecto. O universo inquirido manteve-se constante em ambas as análises e o segundo inquérito foi fechado quando obteve o mesmo número de respostas do anterior. Este facto não significa que exactamente as mesmas pessoas tenham respondido a ambos os inquéritos, mas minimiza a possibilidade de obtenção de respostas de grupos diferentes. Das respostas à avaliação do projecto, mais de metade é proveniente de pessoas que não têm por hábito jogar jogos em dispositivos móveis. Este pode ser um dos motivos pelos quais os resultados foram tão animadores, mas pode também justificar respostas como aquelas que apontam o nível como “quase impossível” ou “amador”. Os jogos para este tipo de equipamentos obedecem muitas vezes a padrões diferentes dos jogos de consolas e apresentam quase sempre gráficos, som e animação de qualidade mais limitada por motivos tão variados como o poder de processamento ou quantidade de memória, os resultados ou preços de venda e o mercado mais pequeno. No entanto, as ideias de bonito ou feio, apropriado ou desadequado obedecem regularmente a certos padrões e os resultados do inquérito não dizem respeito a uma análise de mercado de venda mas sim à qualidade final do trabalho e à viabilidade de projectos caseiros do mesmo tipo.

7.2 Testes

A fase de testes teve como objectivo a resolução de dois tipos de situação. Por um lado a detecção dos erros de desenvolvimento inerentes a um protótipo, e por outro lado a procura de erros conceptuais ou factores que pudessem ser melhorados. Se a utilização de um inquérito respondeu a algumas questões de forma anónima mas geral, um grupo de voluntários permitiu a detecção e comunicação de “bugs” e a partilha de ideias que contribuiriam para uma melhor experiência.

Durante o desenvolvimento do protótipo, testes constantes de jogabilidade deram origem a alterações de vários tipos no jogo, desde correcções de problemas de desenvolvimento até alterações no GDD, com influência directa na experiência de jogo. A exposição diária ao mesmo nível do mesmo jogo criou porém uma certa “resistência” à idealização de novos elementos e à detecção da necessidade de certas alterações. A equipa externa, sem nunca antes ter entrado em contacto com o jogo ou a ideia que o fez nascer, deparou-se com um produto praticamente acabado sem passar pela curva de aprendizagem.

7.2.1 Usabilidade

A primeira situação imprevista surgiu na fase de testes quando o protótipo foi testado na consola GP2X. O primeiro protótipo correu perfeitamente, não apresentando qualquer problema. Quando o protótipo final foi testado na consola, tornou-se bastante óbvio o atraso que o jogo apresentava. Devido à grande quantidade de elementos e aos cálculos necessários para confirmar as colisões entre objectos, o processador da GP2X, com os seus 200MHz, debateu-te quando tentou executar o jogo. Uma das plataformas alvo do projecto não foi suficientemente poderosa para correr o produto final. Apesar de utilizar um processador com a mesma arquitectura, o PocketPC utilizado

para testes, um Eten M600, não apresentou o mesmo problema. Funcionando a uma frequência de 400MHz, o dobro do da GP2X, conseguiu executar o jogo correctamente. A possibilidade de *overclocking* da GP2X poderia ajudar a minimizar este efeito, mas tendo em conta que a velocidade máxima alcançável com a alteração do número de ciclos por segundo varia de exemplar para exemplar, esta consola foi removida do leque de dispositivos em que o jogo deveria correr. A sua descendente, a GP2X Wiz, deverá ser capaz de executar o protótipo com uma fluidez pelo menos idêntica à do telemóvel M600. A falta de uma Wiz para testes deixou esta plataforma de lado até se tornar possível a confirmação das capacidades da consola. Embora o seu processador corra a uma velocidade superior à do do M600, não é possível afirmar sem testes que a Wiz executará a tarefa como se espera. O código do Game-Editor que executa o jogo em diferentes aparelhos pode estar mais ou menos optimizado para cada tipo de equipamento.

No que diz respeito à funcionalidade e experiência de jogo, surgiram duas sugestões relevantes e que sem dúvida fariam parte do jogo final a partir da revisão seguinte. Por um lado foi apontada a necessidade de um compasso de espera no início de cada nível. Embora níveis consecutivos sejam separados por uma imagem informativa, o jogador não sabe exactamente quando deve começar a jogar até que o nível apareça. A introdução de uma contagem regressiva de três para zero após aparecer o ecrã de jogo permitiria ao jogador a preparação para a. Enquanto esta preparação pode parecer desnecessária em alguns tipos de jogo, neste projecto apresenta-se extremamente importante. Devido ao movimento constante do urso há um tempo máximo para ultrapassar cada obstáculo. Este tempo não deve deixar grande margem para transmitir ao jogador a ideia de *suspense* e o obrigar ao movimento constante. Simultaneamente, o intervalo disponível para a resolução do nível é mantido curto para enfatizar o aspecto de *puzzle* do jogo. Se o tempo for muito pode ser possível terminar um nível de múltiplas maneiras, eliminando a necessidade de pensar na estratégia a seguir. A introdução de um período de pausa complementado por uma contagem decrescente no início do nível poderia contornar esta questão, tornando o jogo mais interessante.

Outra questão extremamente pertinente que surgiu da experiência obtida pelos *beta-testers* diz respeito ao desconhecimento do nível que é apresentado pela primeira vez. Completar um nível é uma tarefa fácil para o seu autor, pelo menos no que diz respeito à estratégia de jogo. Cada um dos obstáculos foi cuidadosamente pensado para ser ultrapassado com recurso a determinada técnica, técnica essa que é considerada na criação do nível. Para quem vê este nível pela primeira vez, a disposição dos problemas, ferramentas e objectivos é uma total incógnita e é impossível obter uma visão geral da área de jogo excepto pela utilização da técnica de tentativa e erro. A apresentação de um pequeno mapa transparente seria uma possível solução, mas a baixa resolução disponível cria outros problemas por falta de espaço para a representação do nível. As duas aproximações pensadas para contornar a situação passam pela apresentação da totalidade do nível em algum ponto. Antes do início do nível, a visualização de toda a sua área de forma estática ou deslizando entre o ponto de partida e o objectivo final seria uma possível solução. Outra possibilidade passaria pela introdução de uma nova tecla que serviria para parar temporariamente o jogo e alterar a acção das teclas de controlo. Com o jogo parado, as teclas de movimento deixariam de controlar o castor e passariam a controlar a câmara de forma a permitir a visualização da totalidade do nível. Esta abordagem apresenta o aspecto negativo de simplificar demasiado a resolução do *puzzle*, mas um ponto positivo seria o de possibilitar a sua utilização com níveis de

qualquer tamanho, tanto na horizontal conforme as especificações actuais do jogo, como na vertical, caso esta característica viesse a ser incluída.

7.2.2 “Bugs” e erros de desenvolvimento

Embora as plataformas alvo do jogo sejam as consolas portáteis e alguns telemóveis, o protótipo distribuído foi um ficheiro executável sob o Sistema Operativo Windows da Microsoft. O motivo que levou a esta medida foi o baixo número de consolas portáteis e equipamentos com Windows Mobile disponível entre os *beta-testers*. Esta aproximação não apresenta grandes problemas devido à maneira como o Game-Editor funciona. Apenas o interpretador do Game-Editor varia com o equipamento, enquanto a parte do ficheiro que contém o jogo se mantém constante. O jogo disponibilizado para diferentes sistemas apresenta a mesma informação.

Com o ficheiro executável distribuído num grupo heterogéneo de pessoas, de diferentes idades e sexos e sem conhecimento prévio do jogo, apenas as indicações constantes nas instruções do jogo foram partilhadas. Somente algumas pessoas apontaram a existência de dois “bugs”. O primeiro dizia respeito a uma situação muito comum nos jogos que dá pelo nome “*moonwalking*”. Com o nome alusivo aos passos de dança que tornaram famoso Michael Jackson, quando este se movia para trás embora as suas acções transmitissem a ideia de movimento para a frente, o *moonwalk* acontece quando um personagem de um jogo se move para a esquerda quando virado para a direita ou vice-versa. Esta situação foi tida em conta aquando da criação do protótipo, mas acontece regularmente quando o jogador, sem largar uma das teclas direccionais, prime e larga a tecla na direcção contrária. O lado para o qual está virado o herói controlado pelo jogador é alterado apenas no momento em que a tecla é premida, mas a sua posição é actualizada enquanto a tecla se mantém pressionada. Se não for alterada a animação quando a tecla é largada, o resultado é o efeito descrito acima. Esta situação pode ser presenciada em jogos como “Metal Slug”, na consola NeoGeo.

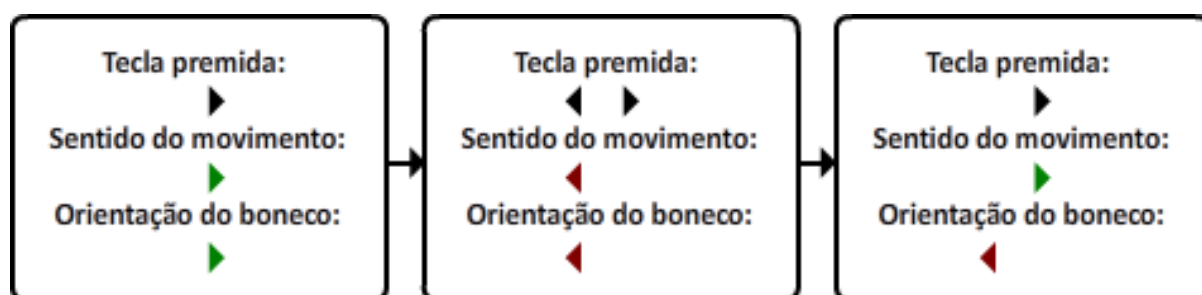


Fig. 81 – Efeito “Moonwalk”

O efeito *moonwalk* foi tido em conta quando o jogo foi desenvolvido e o evento em que a tecla é largada foi monitorizado dando origem à alteração da animação se e apenas se a tecla contrária se mantivesse premida. No entanto, o estado das teclas não estava a ser considerado exactamente no momento inicial do nível. Se a tecla de movimento para a esquerda se encontrasse

premida no início, o sentido do castor não era alterado e este mover-se-ia da direita para a esquerda mas com a cara virada para a direita.

O outro problema detectado passou despercebido à maioria dos elementos da equipa de testes por o considerarem como uma segunda possibilidade de resolução do *puzzle*. Segundo as especificações do jogo, o sonâmbulo acorda quando bate numa parede, resultando na perda de uma vida. Por não ser relevante para a função e para manter a neutralidade, o GDD não foi disponibilizado aos voluntários que testaram o jogo. Quando o urso batia na parede do lado direito da plataforma inferior, continuava a caminhar sem sair do seu lugar, encostado à parede. Sem a pressão do tempo, o jogador podia nesta altura recolher a mola e fazê-la tocar no urso, resultando na projecção deste para a plataforma superior no ponto exacto onde a cama de destino estava colocada. Uma pequena parte do grupo detectou e reportou o erro. O “*bug*” foi corrigido e o jogo passou a ser executado sem problemas visíveis. Das sessenta e duas pessoas que responderam ao inquérito, cinco detectaram problemas. Devido ao anonimato do questionário torna-se impossível a certeza de que mais nenhum “*bug*” foi detectado. Num cenário ideal, o tempo de testes seria alargado de forma a revelar com mais exactidão a isenção de problemas no jogo.

8. Conclusão

Este projecto foi iniciado com o intuito de dar resposta à questão “Será possível a criação de um jogo por um aficionado, como é possível a criação de elementos individuais da matriz dos media?”. A resposta a esta pergunta é simples: sim, é possível. Esta questão foi desde o princípio uma pergunta de retórica, visto que em países como os Estados Unidos da América vários jogos nascem das mãos de equipas extremamente pequenas, muitas vezes com apenas um elemento. O termo “*Indie*”, tantas vezes utilizado ao longo desta dissertação significa exactamente “*Independent Game Developer*”. Jogos como “World of Goo” ou “Toribash” rivalizam com as melhores produções de muitos estúdios especializados e todos os anos surgem inúmeras novidades excepcionais dentro deste tipo de projectos. Então porquê a pergunta? Em Portugal são praticamente desconhecidas as produções caseiras, embora comunidades dedicadas à criação de jogos como o gamedev-pt sejam inundadas por propostas de jogos e pedidos de colaboração. Apesar deste interesse, poucos são os projectos que atingem o estado final e menos ainda são os que chegam a ver a luz do dia. A pergunta assume então a forma “Será possível a criação de um jogo por um aficionado **em Portugal** como é possível a criação de elementos individuais da matriz dos media?”. O protótipo resultante deste trabalho e as respostas aos inquéritos provam que sim, é possível. Será fácil? Não, não o é de forma alguma. O número de competências necessárias é bastante grande, o tempo de desenvolvimento de um jogo completo estende-se por vários meses e é fácil desanimar, especialmente porque é difícil ter apetência para todas as áreas envolvidas.

O protótipo aqui exposto resultou de muito trabalho e ainda assim representa um nível e uma prova de conceito – não um jogo completo. É certo que o gráfico trabalho/tamanho do jogo não apresenta uma curva linear, e um segundo nível necessita apenas de uma fracção do trabalho necessário para criar o primeiro, mas certos elementos representam uma duplicação de trabalho. Algumas animações mantêm-se constantes em múltiplos níveis e em vários mundos do jogo, mas outros componentes, como a música, devem ser desenvolvidos no seu todo sempre que se apresenta uma nova área. Para cada nível um novo LDD deve ser criado e este é talvez o ponto mais importante e o que contribui mais para o baixo número de projectos completos. O planeamento é a fase essencial e a base sobre a qual todos os outros elementos assentam. É fácil ficar empolgado com a ideia de criar um novo jogo, mas se a base não for sólida toda a estrutura corre o risco de ruir. A população americana é muito superior em número à portuguesa, o que justifica logo à partida para uma maior quantidade de projectos de sucesso. Com as grandes indústrias deste mercado situadas maioritariamente nos Estados Unidos, é normal que a cultura dos jogos esteja também mais implantada. Nada do que foi dito invalida a possibilidade de jogos criados em Portugal.

Se muitos jogos *indie* apresentam uma excelente qualidade, estes representam apenas a ponta de um iceberg quando comparados com os jogos mais modestos. Modesto não significa necessariamente “mau”. É importante ter à partida a noção do tempo e quantidade de trabalho que um projecto deste tipo representa, e uma vez mais se volta ao tema do planeamento. A gestão do projecto é uma função pelo menos tão importante como a criação de gráficos ou animações, mas muitas vezes é descurada por ser para muitos menos apelativa. É neste ponto também que é traçada a linha entre os vídeos do Youtube, as fotografias do Flickr ou as músicas do MySpace e a criação de um jogo. Excepto em poucas excepções, os tempos de criação de um vídeo, uma fotografia ou uma

música medem-se em horas ou dias. Os tempos de criação de um jogo medem-se em meses ou anos.

Em cada um dos serviços acima mencionados, é possível encontrar trabalhos de qualidade excepcional, assim como conteúdos francamente maus. Outros ainda revelam a sua vertente amadora mantendo o padrão de qualidade bastante aceitável. A criação de um clone do jogo Tic-Tac-Toe na sua totalidade teria certamente demorado menos tempo do que apenas a componente gráfica do jogo Sleepwalker. Isto não significa que ao iniciar a criação de um jogo se deve começar com baixos padrões, mas sim que o tempo e o trabalho devem ser analisados e partir para o projecto apenas após confirmar a sua viabilidade.

Outra preocupação deste trabalho foi o desenvolvimento de um conceito de um jogo original e inovador. Os jogos atrás referidos, “World of Goo” e “Toribash” são exemplos de originalidade num mercado repleto de ideias e provam que há ainda muito para criar. Sem a aspiração de alcançar os níveis de originalidade dos jogos mencionados, com base nas respostas ao inquérito de avaliação do protótipo é possível dizer que a ideia é de facto original.

Durante o desenvolvimento desta tese várias novidades foram reveladas e novos produtos lançados. A Pandora, que no início deste trabalho estava a alguns meses do seu lançamento, deverá ser distribuída muito brevemente, e a Sony PSP Go deverá aparecer nos mercados europeu e americano no dia 1 de Outubro, e no mercado Japonês um mês depois. Ainda antes do lançamento da Pandora, o sucessor do seu processador – o Cortex A9 *multiprocessor* – foi já anunciado. O mercado do *hardware* continua a sua constante evolução a um ritmo acelerado, abrindo múltiplas possibilidades para *software* e novos conceitos.

O Game-Editor foi escolhido para o desenvolvimento deste projecto não por ser considerado o melhor na sua gama, mas por permitir a criação de jogos para plataformas móveis de uma forma relativamente simples. Em termos de engenharia de *software*, o desenvolvimento de uma aplicação que permita a exportação de jogos para a nova consola OpenPandora revela-se um excelente tema de trabalho. Em Outubro de 2008, o código fonte do Game-Editor foi disponibilizado. Apesar de não ser exactamente distribuída como *Open Source*, a aplicação está aberta a qualquer pessoa que deseje contribuir para o projecto, bastando para isso enviar uma mensagem ao autor, Makslane Rodrigues. Alternativamente, o desenvolvimento de raiz de uma aplicação similar poderia dar origem a um produto paralelo. O Game-Editor sofre de algumas lacunas, especialmente no que diz respeito ao interface, e um produto que disponibilizasse uma forma mais intuitiva e organizada de manipular as propriedades dos objectos em jogo teria com certeza grande aceitação. Um método de trabalho similar ao utilizado pelos programas MaxMSP ou PureData poderia oferecer ao mundo dos jogos para plataformas móveis aquilo que estas aplicações trouxeram ao universo das instalações multimédia.

Ainda no ponto do *software*, e com os novos equipamentos de bolso a apresentar desempenhos que começam a aproximar-se dos *netbooks*, uma aplicação que permita a criação de jogos na própria Pandora ou terminal similar poderia certamente revelar-se uma excelente ferramenta.

No que diz respeito a novos conceitos de jogo, o poder de processamento que é presentemente oferecido numa consola de bolso introduz uma panóplia de possibilidades até agora

inexistente. A oferta de gráficos em três dimensões com aceleração permite actualmente dotar de uma nova vida ideias antigas que servem de base a jogos como “The Incredible Machine”, jogo em que o objectivo assenta na criação de máquinas ao estilo de Rube Goldberg, publicado pela Sierra Entertainment. Simultaneamente, a utilização das normas *USB Host* e *USB-On-The-Go* oferecida pelas novas plataformas adequa-se à introdução de novos métodos de controlo, como acelerómetros, GPS, bússolas ou mesmo a integração de plataformas como as placas Arduino e similares, abrindo caminho para novos conceitos de interacção. Empresas como a Nintendo canalizam grandes quantidades de recursos humanos e financeiros para os seus departamentos de Investigação e Desenvolvimento, o que prova que existe uma grande procura da pesquisa nestas áreas.

Haverá bons motivos para despender de tanto tempo na criação de um jogo para dispositivos móveis? O mercado deste tipo de equipamentos está em crescimento, com empresas como a HTC a lançar novos modelos regularmente, a Apple a apresentar novas revisões do seu iPhone com curtos intervalos e terminais com Android a surgir constantemente de vários fabricantes. Os MIDs (*Mobile Internet Devices*) começam a ganhar mercado e todo este tipo de equipamentos caminha em direcção à maturidade. Programadores independentes começam pequenas fortunas a vender as suas aplicações a \$0.99 em *AppStores*. A globalização permite a qualquer pessoa, em qualquer parte do mundo, disponibilizar o seu produto em todas as áreas do planeta.

É possível desenvolver um jogo em casa, original, para dispositivos móveis e em Portugal, e há muitos e bons motivos para o fazer, como a aprendizagem, a divulgação do produto final ou o eventual retorno monetário que advém da venda do jogo. No fim de um projecto deste tipo não se vê um castor, vê-se o Inácio. Não se vê um urso, vê-se o Godofredo. Não se vê um jogo, vê-se o Sleepwalker. Vê-se o sonho de um urso e a amizade de um castor e deseja-se-lhes sucesso.



Referências Bibliográficas

- Albertarelli, Spartaco. 2000. 1000s Ways to Play Monopoly. *Board Game Studies* (3):149.
- Ali, Daemon. 2009. *Senet, Weiqi, and the World's Oldest Still Played Board Games*. World Time News Report 2007 [cited January 24 2009]. Available from <http://www.wtnradio.com/story.php?story=185>.
- AllExperts.com. 2009. *Gizmondo*. AllExperts.com [cited February 2009]. Available from <http://en.allexperts.com/e/g/gi/gizmondo.htm>.
- Amaro, Pedro. 2009. *The Clash of Mobile Platforms: J2ME, ExEn, Mophun and WGE*. Gamedev.net 2003 [cited February 2009]. Available from <http://www.gamedev.net/reference/articles/article1944.asp>.
- Betabreakers. Available from <http://www.betabreakers.com/>.
- Boardgamegeek. 2009. *Vinícula*. Boardgamegeek 2009 [cited January 24 2009]. Available from <http://www.boardgamegeek.com/boardgame/42052>.
- Boffi, Guido, and Artur Morão. 1999. <<Os >>caminhos do jazz, *Convite à música* 6. Lisboa: Edições 70.
- Byous, Jon. 2009. *Java Technology: The Early Years* 2003 [cited February 2009]. Available from <http://java.sun.com/features/1998/05/birthday.html>.
- CaliforniaMilkProcessorBoard. 2009. *Get The Glass*. California Milk Processor Board 2007 [cited January 25 2009]. Available from <http://www.gettheglass.com/>.
- Cantor, Doug. 2006. GAME IT YOUR WAY. *Popular Science* 269 (5):100-100.
- Chikun, Cho, ed. 1997. *Go - A complete introduction to the game*: Kiseido Publishing Company.
- CSPProjects. 2009. *Main Game Loop Architecture* [cited March 2009]. Available from <https://cs.fit.edu/proxy/proj/SidusBellum/wiki/Main%20Game%20Loop%20Architecture>.
- Cunha, Luís Veiga da. 1999. *Desenho técnico*. 11ª ed ed, *Manuais universitários*. Lisboa: Fundação Calouste Gulbenkian.
- Deffree, Suzanne. 2008. Nokia buys Symbian shares; teams with TI, ST, others for open platform Symbian Foundation. *Electronic News* (10616624) 54 (26):10.
- DeMaria, Rusel, and Johnny L. Wilson. 2003. *High Score! The Illustrated History of Electronic Games* 2nd ed: McGraw-Hill Companies.
- EdgeLib. 2009. *EdgeLibe*. EdgeLib 2005 [cited February 2009]. Available from <http://www.edgelib.com/index.php?node=license>.
- Ferreira, Pedro, and Emília Azevedo. 2001. *Flash <<5=5>> conceitos e prática*. 2ª ed ed. Lisboa: FCA - Editora Informática.
- Gartner. 2009. Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008.

- GPH. 2009. *Gamepark Holdings*. Gamepark Holdings Co., Ltd. 2008 [cited February 2009]. Available from <http://www.globalgph.com/>
<http://www.gp2x.com>.
- Grand, Joe, Albert Yarusso, Ralph H. Baer, Marcus R. Brown, and Frank Thornton. 2004. *Game Console Hacking: Xbox, PlayStation, Nintendo, Game Boy, Atari, & Sega*: Andrew Williams.
- Gravelyn, Nick. 2009. *Animating Sprite Sheets And Moving Sprites*. Ziggyware 2007 [cited March 2009]. Available from http://www.ziggyware.com/readarticle.php?article_id=138.
- Handheldmuseum. 2009. *Cragstan Periscope-Firing Range*. handheldmuseum.com 2005 [cited January 25 2009]. Available from <http://www.handheldmuseum.com/Misc/CragstanPeriscope.htm>.
- Hashim, H. A., S. H. Ab Hamed, and M. E. M. Sabri. 2007. *Analysis on types of mobile games played among the IHL students in Malaysia, PROCEEDINGS OF THE 6TH WSEAS INTERNATIONAL CONFERENCE ON EDUCATION AND EDUCATIONAL TECHNOLOGY (EDU'07)*.
- Huizinga, Johan. 1970. *Homo Ludens: a study of the play element in culture*. London,: Maurice Temple Smith Ltd.
- MidletReview. 2009. *Major N-Gage title 'Worms World Party' created using mophun*. MidletReview 20042009]. Available from <http://www.midlet-review.com/index?content=news&id=604>.
- MultitouchBarcelona. 2009. Multitouch Barcelona [cited November 2nd 2009]. Available from <http://www.multitouch-barcelona.com/>.
- Nintendo. 2005. Nintendo Annual Report. Kyoto.
- . 2008. Consolidated Financial Highlights. Kyoto: Nintendo Co., Ltd.
- Nix. 2009. *A New Taste of the PSP Chips*. ign.com 2005 [cited February 2009]. Available from <http://psp.ign.com/articles/542/542182p1.html>.
- OpenPandora. 2009. *OpenPandora.org*. OpenPandora.org 20072009]. Available from <http://www.openpandora.org>.
- Parrish, Kevin. 2009. *PSP Go is 43% Lighter, No UMD* 2009 [cited June 2009]. Available from <http://www.tomsguide.com/us/PSP-Go-Sony-Handheld,news-4025.html>.
- Pulli, Kari, Tomi Aarnio, Ville Miettinen, and Kimmo Roimela. 2007. *Mobile 3D Graphics: with OpenGL ES and M3G*: Morgan Kaufmann.
- Ribeiro, Nuno. 2004. *Multimédia e tecnologias interactivas, Tecnologias de informação*. Lisboa: FCA - Editora de Informática.
- Rodrigues, Alvaro Gabriele. 2009. Game Design. Slideshare.
- Rodrigues, Makslane. 2009. *Game-Editor*. Makslane Rodrigues 2003 [cited January 2009]. Available from <http://game-editor.com>.

- Ryan, Tim. 2009. *The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal*. Gamasutra 1999 [cited March 2009]. Available from http://www.gamasutra.com/view/feature/3384/the_anatomy_of_a_design_document.php.
- Sloper, Tom. 2009. *Testers - The Unsung Heroes of Games*. Sloperama 2001 [cited April 2009]. Available from <http://www.sloperama.com/advice/lesson5.htm>.
- Small, Mike. 2009. *Get a Job Video Game Testing - Quality Assurance for Electronic Arts*. Orato 2008 [cited April 2009]. Available from <http://www.orato.com/tech-games/get-job-video-game-testing>.
- Sony. 2005. PLAYSTATION®2 BREAKS RECORD AS THE FASTEST COMPUTER ENTERTAINMENT PLATFORM TO REACH CUMULATIVE SHIPMENT OF 100 MILLION UNITS. Tokio.
- Soubeyrand, Catherine. 2009. *The Royal Game of Ur*. The Game Cabinet 1996 [cited January 25 2009]. Available from <http://www.gamecabinet.com/history/Ur.html>.
- Thegameconsole.com. 2009. *Video Game Consoles*. The Game Console 2006 [cited January 28 2009]. Available from <http://www.thegameconsole.com/>.
- Thompson, Jim, Barnaby Berbank-Green, and Nic Cusworth. 2007. *The Computer Game Design Course*. London: Thames & Hudson.
- Videogametesterhub. 2009. [cited April 2009]. Available from <http://www.videogametesterhub.com/top-video-game-tester-companies.html>.
- Wayne, Rick. 2006. Flash Lite 2.0. *Dr. Dobb's Journal: The World of Software Development* 31 (6):42.

ANEXO 1 – Game Design Document

Conceito: Hugo Machado

Data: Março de 2009

Título: Sleepwalker

Subtítulo: As noites atribuladas de Godofredo

Versão: 1.0

Género: *Puzzle*/Plataformas

Idade alvo: 15-40 anos

Plataformas: Gamepark Holdings GP2X Wiz / Windows Mobile

Conceito

O urso Godofredo tem um sono irrequieto. As suas preocupações não o deixam dormir descansado e, sem acordar, levanta-se durante a noite à procura do mel dos seus sonhos. Como seu melhor amigo, o castor Inácio zela para que nada de mau lhe aconteça.

Sinopse

Após umas férias de verão na terra do primo Bonifácio, o urso Godofredo volta a casa com saudades das colmeias abundantes em mel que conheceu durante a sua estadia.

De volta a casa, Godofredo sonha constantemente com o mel. Os sonhos das férias não o deixam descansar e, na busca incessante pela iguaria levanta-se noite após noite da sua cama, durante o sono, à procura do almejado prémio.

O castor Inácio, amigo próximo de Godofredo e seu vizinho, alertado para os perigos a que está sujeito o seu camarada, passa as noites acordado a zelar pela vida e segurança de Godofredo.

Elementos e características do jogo

Sleepwalker é um jogo de plataformas com elementos de *puzzles*. O herói Inácio, controlado pelo jogador, tem como função a protecção de Godofredo.

Godofredo é um misto de herói/NPC, visto que o jogo se desenvolve à sua volta mas este não é controlado pelo jogador. Mantém um movimento constante até encontrar algum objecto que altere a sua trajectória. Os objectos são manipulados por Inácio. A função de Inácio é permitir uma trajectória segura para Godofredo desde uma cama de início até uma cama de fim do nível.

O jogo é constituído por vários níveis, passados em mundos diferentes. Em cada nível, Godofredo levanta-se de uma cama durante o sono. O nível é completado com sucesso sempre que o urso Godofredo alcance a cama de destino sem ter entretanto acordado.

Ao longo de cada nível estão disponíveis objectos que permitem a Inácio alterar os obstáculos de forma a permitir a passagem segura de Godofredo de uma cama para outra.

Em cada nível estão espalhadas maçãs que Inácio pode recolher como objectivos secundários. Estas maçãs não só incrementam a pontuação do jogador como permitem a Inácio deslocar-se mais rapidamente, devido à energia fornecida. Em certos casos são necessárias para permitir a resolução do *puzzle* em tempo útil.

O movimento do urso Godofredo tem uma velocidade constante. Esta velocidade pode ser alterada por variações do terreno, naturais ou produzidas pela interacção de Inácio.

Mercado alvo

Sleepwalker é um jogo destinado à faixa etária dos 15 aos 40 anos. O jogo exige raciocínios elaborados para a solução dos níveis, não sendo portanto destinado ao público mais jovem.

Tendo como plataformas alvo as consolas abertas e telemóveis de gama alta com Windows Mobile, os potenciais jogadores são jovens adultos com o dinheiro necessário para comprar telemóveis de topo de gama com funções profissionais e aficionados de Linux, existente nas consolas a que se destina.

Com elementos de jogo originais, Sleepwalker aproxima-se do estilo de jogo de “Lemmings” ou “Lost Vikings”, sucessos conhecidos da geração de 70. Contém também elementos visuais e de jogabilidade que poderão atrair um público mais jovem.

Com níveis curtos e a possibilidade de gravar o jogo após cada nível, o jogo pode ser jogado nos transportes públicos ou numa fila para serviços públicos, atraindo pessoas com pequenos intervalos de tempo livre.

Arte – Aspecto Visual

Visualmente o jogo apresenta um visual simples e colorido. Com uma vista lateral bidimensional da acção, os elementos são visualizados de lado ou com uma pequena inclinação na direcção do observador.

Os vários mundos jogáveis representam florestas, grutas, desertos e quintas com os seus elementos característicos a utilizar uma paleta de cores completa. Os fundos apresentam tonalidades de cinzento ou azul escuro visto que a acção se desenrola de noite. Os últimos níveis de cada área podem ser representados com as cores mais saturadas e brilhantes características do nascer do dia.

Todos os elementos utilizam linhas simples similares às de desenhos animados.

Quatro níveis de profundidade aparente fornecem a ilusão de uma terceira dimensão, começando com o céu estrelado e terminando no plano onde se desenrola a acção.

Objectivo do jogo

O objectivo principal de Sleepwalker é facilitar o movimento do urso até ao nível final, um paraíso cheio de colmeias e de mel.

Objectivos secundários

No desenrolar dos níveis, o castor pode optar por recolher mais ou menos maçãs. Estas maçãs proporcionam um aumento súbito de energia que permite ao castor uma maior velocidade em relação à do urso ou dos inimigos. A recolha das maçãs pode ser facultativa ou obrigatória para a conclusão do nível.

Mecânica do jogo

O jogo é constituído por várias plataformas a alturas diferentes.

O urso segue um movimento constante na horizontal até este movimento ser alterado. No percurso inicial do urso estão distribuídos diferentes obstáculos que dificultam ou impedem o trajecto.

O castor é controlado pelo jogador e pode movimentar-se para cima e para baixo, para a esquerda e para a direita. O castor recolhe objectos automaticamente sempre que lhe toca. Sempre que transporta um objecto quando toca em outro, o objecto original é largado de forma a permitir o transporte do novo. É possível também largar um objecto propositadamente, sem apanhar um novo objecto, com a utilização de uma tecla dedicada. O objecto corrente pode ser utilizado automaticamente sempre que surge uma situação relevante ou manualmente. Alguns objectos, como armas, devem ser accionados por meio de uma tecla. É permitido ao castor subir e descer sempre que se encontre por cima de uma escada. É ainda permitido saltar curtas distâncias.

Menus

Ecrã inicial

- Novo Jogo
- Continuar
- Carregar / Gravar Jogo
- Instruções
- Sair

Novo Jogo – Começa um jogo desde o início

Continuar – Continua o jogo interrompido por meio do menu de opções no jogo.

Carregar / Gravar Jogo

- Carregar Jogo – Carrega um jogo previamente gravado.
- Gravar Jogo – Grava o jogo actual.
- Voltar – Volta ao menu principal

Instruções – Apresenta as instruções de jogo.

Sair – Abandona o jogo

Durante a execução do jogo é possível a sua interrupção para a apresentação de um menu com as seguintes opções:

- Volume de som – Visualiza uma barra horizontal cujo comprimento reflecte o volume dos efeitos sonoros. Esta barra pode ser controlada com as teclas.
- Volume da música – Visualiza uma barra horizontal cujo comprimento reflecte o volume da música. Esta barra pode ser controlada com as teclas.
- Voltar ao jogo – Volta ao jogo em execução.
- Voltar ao menu – Volta ao menu principal.

Controlos

O castor é controlado por meio das teclas direccionais. É-lhe permitida a deslocação na vertical apenas quando se encontra sobre uma escada.

Consolas:

◀ - ▶ - Movimento do castor. Esquerda ou direita.

▲ - ▼ - Movimento do castor. Cima e Baixo. Apenas quando o castor está posicionado sobre uma escada.

A – Larga ou dispara o objecto transportado (se não for um objecto utilizado automaticamente).

B – Salto do castor.

Start – Apresenta o menu das opções.

Windows Mobile:

◀ - ▶ - Movimento do castor. Esquerda ou direita.

▲ - ▼ - Movimento do castor. Cima e Baixo. Apenas quando o castor está posicionado sobre uma escada.

Soft Key 1 – Larga ou dispara o objecto transportado (se não for um objecto utilizado automaticamente).

Soft Key 2 – Salto do castor.

Action – Apresenta o menu das opções.

As teclas *Soft Key 1* e *Soft Key 2* podem ser redefinidas. As suas acções podem também ser executadas pela utilização de botões virtuais respectivamente nas áreas inferior e superior do ecrã.

Informação no ecrã

Canto superior esquerdo: Seta que aponta a posição do urso. Pode rodar 360° sobre si mesma. Um valor numérico ao lado da seta indica a distância a que o urso se encontra do castor. Este valor varia entre 0 e 100, sendo que para distâncias superiores ao valor máximo são representadas igualmente pelo valor 100.

Canto superior direito: Pontuação – valor numérico.

Canto inferior esquerdo: Número de vidas – pequenas cabeças de urso na quantidade representante do número de vidas disponível, organizadas numa fila horizontal e equidistantes umas das outras.

Vida e energia dos personagens

O jogador perde uma vida sempre que:

- O castor caia dentro de água ou entre em contacto com um inimigo.
- O urso caia de um lugar alto, caia dentro de água, entre em contacto com um inimigo ou acorde do seu sono antes de chegar a uma cama.

As maçãs recolhidas pelo castor aumentam temporariamente a velocidade do seu movimento, que diminui gradualmente sempre que o seu valor seja superior ao mínimo (velocidade no início do nível).

As velocidades do castor e do urso podem também sofrer alterações para cima ou para baixo conforme o tipo de terreno.

Todos os outros personagens móveis apresentam velocidades constantes.

Câmaras

O castor é sempre o alvo da câmara. Ocupa sempre uma posição nos 33% do espaço central do ecrã e todos os elementos são posicionados à sua volta. É sempre visto de lado, assim como os restantes elementos no ecrã. Alguns objectos apresentam uma ligeira inclinação sobre o eixo vertical que contribui para uma sensação de profundidades, assim como os quatro níveis de “Paralaxe”.

Os vários constituintes do jogo movem-se em relação ao solo e nenhum (exceptuando o castor) está constantemente visível. Para que o jogador possa localizar o urso, uma seta no canto superior esquerdo do ecrã aponta a posição deste, podendo rodar 360º sobre si mesma. Ao lado da seta, um valor numérico entre 0 e 100 indica a distância a que o urso se encontra do castor.

Vários níveis de terreno podem existir no mesmo nível e podem estar ligados por escadas ou rampas. O urso não pode subir nem descer escadas.

Áreas do jogo

O jogo é constituído por quatro mundos: floresta, deserto, gruta e quinta (ou pomar). Cada mundo ou área é constituído por dez níveis com alusões temáticas ao ambiente em que se enquadram.

A mecânica de jogo é linear, sendo o objectivo permitir a deslocação do urso entre o ponto A e o ponto B representados por camas onde o urso repousa entre níveis. A estratégia de jogo é variável, podendo o castor resolver primeiro o *puzzle* 1 e depois o *puzzle* 2 ou vice-versa. Em certas situações podem existir várias soluções para um mesmo nível.

Física do jogo

Todos os personagens do jogo são afectados pela gravidade, mantendo-se sobre a plataforma em que se encontram, com excepção dos personagens que voam. Estes seguem caminhos pré-estabelecidos.

As maçãs estão suspensas no ar para serem facilmente identificadas. Todos os outros elementos inanimados são afectados pela gravidade.

Ao castor são permitidos saltos de curta distância. Estes saltos partem de uma aceleração vertical constante e podem atingir diferentes distâncias na horizontal conforme a velocidade do castor no momento do salto.

Lama e água de baixa profundidade diminuem a velocidade do urso e do castor. Esteiras rolantes alteram a direcção do urso se os seus movimentos forem contrários e reduzem a velocidade do castor. Se o movimento das esteiras for no mesmo sentido do movimento do urso ou do castor, estes sofrem uma aceleração. Esta redução ou ampliação da velocidade mantém-se apenas enquanto o personagem caminhar sobre a esteira.

Ao atingir um ponto que não apresente uma superfície sob os pés de ambos os heróis do jogo, como o fim de uma plataforma, ambos caem na plataforma inferior por acção da gravidade. O urso acorda com a queda, resultando na perda de uma vida.

Todos os personagens animados à excepção do urso e do castor resultam na perda de uma vida quando tocados. Com a utilização dos objectos relevantes, estes personagens ou elementos antagonistas podem ser destruídos.

Outros personagens

Personagens animados podem aparecer em qualquer nível e variam com o mundo em que se encontram.

Morcego (Floresta e Gruta) – Voa em movimento circular pré-estabelecido. Um toque em qualquer um dos heróis é fatal. Pode ser destruído com a fisga (manual,

longo alcance) a pedra (manual, alcance médio) ou com o pau (automático, curta distância).

Cobra (Floresta e Deserto) – Arrasta-se pelo chão de um lado para o outro a uma curta distância. Um toque em qualquer um dos heróis é fatal. Pode ser destruída com a fisga (manual, longo alcance) a pedra (manual, alcance médio) ou com o pau (automático, curta distância).

Abutre (Deserto) – Voa em movimentos longos e rápidos pré-estabelecidos. Um toque em qualquer um dos heróis é fatal. Pode ser destruído com a fisga (manual, longo alcance) a pedra (manual, alcance médio) ou com pau (automático, curta distância).

Aranha (Gruta e Pomar) – Desloca-se verticalmente para cima e para baixo sobre a sua teia. Um toque em qualquer um dos heróis é fatal. Pode ser destruída com a fisga (manual, longo alcance) a pedra (manual, alcance médio) ou com o pau (automático, curta distância).

Ouriço (Floresta e Pomar) - Arrasta-se pelo chão de um lado para o outro a uma curta distância, mais rapidamente do que a cobra. Um toque em qualquer um dos heróis é fatal. Pode ser destruída com a fisga (manual, longo alcance) a pedra (manual, alcance médio) ou com o pau (automático, curta distância).

Planta Carnívora (Deserto) – Não se desloca nem pode ser destruída, deve ser evitada. Um toque em qualquer um dos heróis é fatal.

Piranha (Floresta) – Salta da água. Um toque em qualquer um dos heróis é fatal. Não pode ser destruída mas pode ser temporariamente atordoada com a pedra (manual, alcance médio) ou com a fisga (manual, longo alcance) enquanto se encontra fora da água.

Corvos (Pomar) – Deslocam-se em grupo. Um toque em qualquer um dos heróis é fatal. Podem ser destruídos, um a um, com a fisga (manual, longo alcance) a pedra (manual, alcance médio) ou com pau (automático, curta distância).

Espantalho (Pomar) - Não se desloca nem pode ser destruído, deve ser evitada. Não provoca alterações no castor. A proximidade do urso faz com que este acorde assustado, resultando na perda de uma vida.

Outros elementos inanimados

Vidros – Em qualquer dos mundos podem aparecer espalhados no chão. Um toque do urso resulta na perda de uma vida. Não produz efeito no castor. Se o castor tiver uma vassoura, automaticamente varre os vidros que desaparecem, tornando o percurso seguro para o urso.

Fogo – Em qualquer dos mundos pode aparecer no chão. Um toque em qualquer dos heróis resulta na perda de uma vida. Se o castor tiver um regador seleccionado, automaticamente apaga o fogo tornando o percurso seguro.

Água – Em qualquer dos mundos pode aparecer no chão. Se a profundidade for pouca, atrasa o movimento de ambos os heróis. Se for muita, os heróis afogam-se, resultando na perda de uma vida. Pode conter piranhas que devem ser atordoadas com uma pedra sempre que se encontrem fora de água. Pode ser atravessada se o castor distribuir câmaras-de-ar na sua superfície.

Armas, equipamentos e outros objectos

Fisga – Pode ser utilizada pelo castor manualmente para atirar pedras aos inimigos eliminando-os (ou atordoando no caso da piranha). É eficaz a longas distâncias. Só pode ser utilizada uma vez.

Pau – Pode ser utilizado pelo castor automaticamente para eliminar inimigos. É eficaz a curta distância.

Pedra – Pode ser utilizada pelo castor manualmente para atordoar a piranha ou eliminar os restantes inimigos. É eficaz a médias distâncias. Só pode ser utilizada uma vez.

Vassoura – Pode ser utilizada pelo castor automaticamente para varrer os vidros que se encontram pelo caminho.

Regador – Pode ser utilizado pelo castor automaticamente para apagar o fogo que aparece pelo caminho.

Câmaras-de-ar – Podem ser largadas pelo castor sobre a água para criar uma ponte.

Esteiras rolantes – Aceleram o movimento de ambos os heróis se a sua direcção for igual à do herói. Se a direcção do movimento for contrária à do herói, desacelera o movimento do castor e inverte o sentido do movimento do urso.

Maças – Quando recolhidas pelo castor aceleram temporariamente a sua velocidade.

Mola – Pode ser movida pelo castor e largada em qualquer ponto. Projecta o urso para a plataforma superior àquela em que este se encontra, caso esta exista.

Almofada – Deve ser posicionada sob o fim de uma plataforma elevada para que o urso possa continuar o seu caminho após uma queda.

Animações

Personagens

Urso

- Vistas laterais esquerda e direita, a caminhar.
- Vista frontal a abanar os braços.

Castor

- Vistas laterais esquerda e direita, a caminhar.
- Vista frontal a abanar os braços.
- Vista traseira a subir escadas.
- Vistas laterais esquerda e direita a disparar a fisga.
- Vistas laterais esquerda e direita a usar o pau.
- Vistas laterais esquerda e direita a atirar a pedra.

Morcego

- Vistas laterais esquerda e direita, em voo.

Cobra

- Vistas laterais esquerda e direita, a arrastar-se pelo chão.

Abutre

- Vistas laterais esquerda e direita, em voo.

Aranha

- Vista frontal, a subir e descer a teia. A teia é incluída na animação.

Ouriço

- Vistas laterais esquerda e direita, a caminhar.

Planta Carnívora

- Vistas laterais esquerda e direita da planta a abrir e fechar a “boca”.

Piranha

- Vistas laterais esquerda e direita da piranha a nadar.
- Vistas laterais esquerda e direita da piranha a abrir e fechar a boca.

Corvos

- Vistas laterais esquerda e direita a voar.

Espantalho

- Vistas laterais esquerda e direita, estática.
- Vistas laterais esquerda e direita a assustar o urso.

Obstáculos

Vidros

- Pedacos de vidro no chão.

Fogo

- Chamas.

Água

- Ondulação na superfície da água.

Armas, equipamentos e outros objectos

Vassoura – Vistas laterais esquerda e direita, nas mãos do castor.

Regador – Vistas laterais esquerda e direita com água a sair.

Câmaras-de-ar – Vista lateral a abanar com a ondulação da água.

Esteiras rolantes – Vistas laterais esquerda e direita a “rolar”.

Maçãs – Vista frontal a oscilar.

Mola – Vista frontal a comprimir e distender.

Almofada - Vista frontal a comprimir.

Guião do jogo

Com sonhos constantes sobre o mel facilmente acessível que viu na casa do primo, o urso Godofredo levanta-se da cama todas as noites e começa a caminhar sem acordar. O seu melhor amigo, o castor Inácio, zela pela sua segurança, ciente de que não é boa ideia acordar um sonâmbulo.

A persistência de Godofredo leva-o a não desistir da sua busca, ainda que inconsciente dela, até encontrar o almejado mel.

Na sua epopeia, Godofredo sairá da sua casa na floresta e caminhará sem consciência por diferentes ambientes. Inácio seguirá o seu amigo protegendo-os dos diferentes perigos a que um sonâmbulo está constantemente sujeito.

Atravessando toda a floresta, onde Godofredo nasceu e viveu toda a sua vida, o urso pára apenas para dormir uns tempos descansado quando encontra uma cama. Mas o seu descanso, assim como o do amigo Inácio, duram geralmente muito pouco tempo, porque o subconsciente do urso geralmente prevalece sobre o seu cansaço.

A floresta termina no início de um longo deserto, onde os dois amigos enfrentam novos perigos.

No fim do deserto, uma gruta por entre as montanhas apresenta novas dificuldades.

Do outro lado da montanha, após atravessar a gruta, um pomar cheio de colmeias e mel aguarda finalmente pelo duo. Para chegarem ao mel terão no entanto que ultrapassar os diversos perigos que se encontram nas terras da fruta.

Níveis

Em anexo no **Level Design Document**.

Programação

Motor de jogo - Game Editor

Música e Som

- Música no ecrã de entrada, tema principal.
- Música nos níveis, um tema diferente por cada mundo.

O jogo utiliza harmonias simples diatónicas e melodias fáceis de entrar no ouvido. O tipo de sons usados é alusivo ao mundo em que a acção decorre.

Efeitos sonoros para:

- Apanhar maçã.
- Utilizar arma ou um objecto.
- Arma atingir inimigo.
- Castor apanhar maçãs.
- Mola a distender.
- Queda sobre a almofada.
- Vassoura a varrer vidros.
- Água a apagar fogo.
- Planta carnívora ou piranha a bater os dentes.
- Voz aguda de aproximadamente 2 segundos quando se perde uma vida.
- Voz aguda de aproximadamente 2 segundos quando se termina o nível.

ANEXO 2 – Level Design Document

Jogo: Sleepwalker – As noites atribuladas de Godofredo

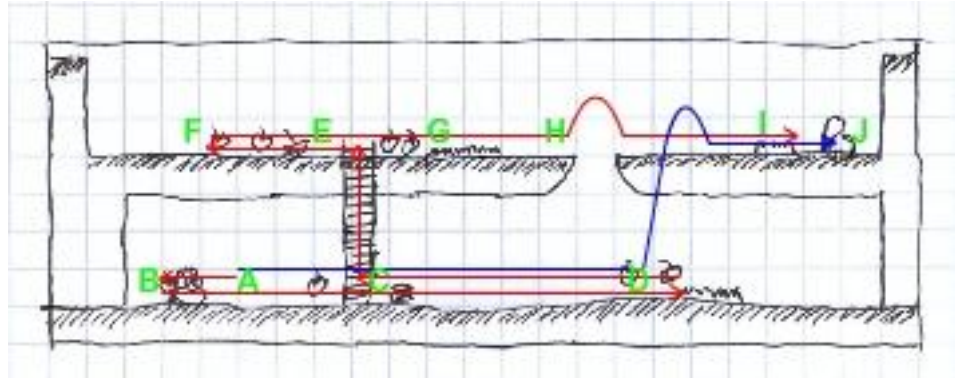
Área: Floresta

Nível: 10

Versão: 2.0

Área: Floresta

Nível 10



Tempo estimado para resolver o puzzle: 5 minutos

Tempo estimado para completar o nível após a resolução do puzzle: 30 segundos

Descrição

Este nível representa a última secção da Floresta. Ambos os heróis iniciam o nível à direita da cama posicionada do lado esquerdo da plataforma inferior, no ponto A.

Não existem inimigos móveis. As três condições para Godofredo atingir o seu destino são:

- 1 – Passar à plataforma superior
- 2 – Não cair na plataforma inferior devido a uma queda.
- 3 – Não pisar os vidros.

Solução

A Velocidade

Partindo do ponto A, o jogador deve dirigir-se ao ponto B e apanhar as maçãs à sua esquerda. A velocidade conseguida com as duas maçãs contribui para a vantagem necessária para completar o nível.

A Mola

No ponto **C** o castor apanha a mola que deve ser largada no ponto **D**, apanhando as três maçãs pelo caminho. Ao chegar a este ponto, o urso será projectado para a plataforma superior.

A Vassoura

Voltando rapidamente para trás e após subir a escada, o jogador apanha a vassoura no ponto **E**, que lhe permitirá mais tarde limpar os vidros à direita da plataforma superior.

A Velocidade, parte 2

O jogador deve agora recolher as maçãs posicionadas à sua esquerda, no ponto **F**, visto que o pequeno atraso resultante de uma breve caminhada no sentido contrário ao do seu objectivo será compensado pelo acréscimo de velocidade resultante da recolha das maçãs.

Os Vidros

No ponto **G** os vidros serão limpos com a vassoura. Estes vidros não influenciam a resolução do nível porque o urso aparece na plataforma superior apenas à direita destes.

O Salto

Em **H** o jogador deve premir a tecla de salto de forma a ultrapassar a falha entre plataformas. Com as maçãs recolhidas este salto não será neste momento complicado porque o castor move-se neste momento com uma velocidade horizontal bastante superior à necessária para ultrapassar o obstáculo.

Os Vidros, parte 2

Os vidros em **I** têm que ser obrigatoriamente varridos antes de o urso lhes tocar. Se o jogador tiver recolhido todas as maçãs e não tiver parado durante o nível, chegará a este ponto antes do urso e, munido da vassoura, limpará o chão.

O descanso

Com todos os obstáculos ultrapassados, o urso deverá chegar à cama no ponto **J** sem qualquer precalço.

Sleepwalker

Fim da Floresta

Entrada para a *cut-scene* relativa ao deserto.

Anexo 3 – Arte Conceptual



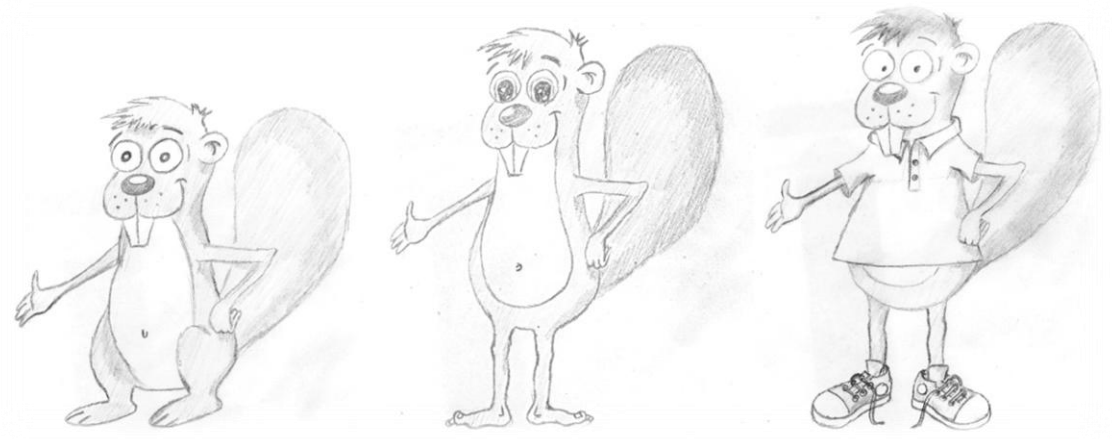
Casa do Urso Godofredo



Godofredo



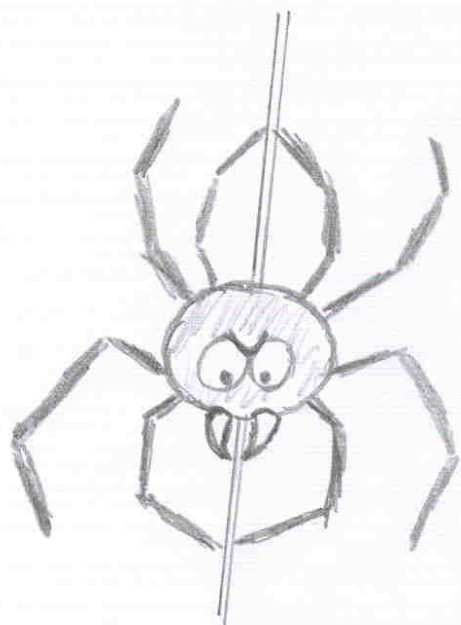
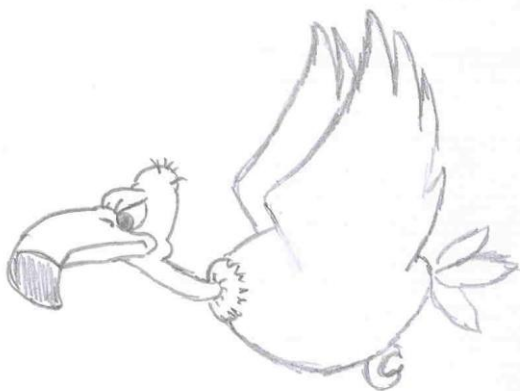
Inácio – Ideia inicial



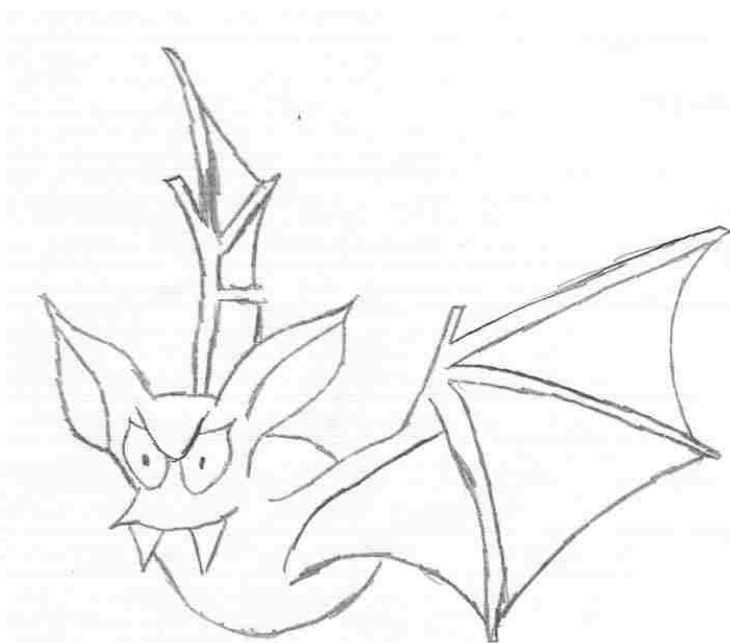
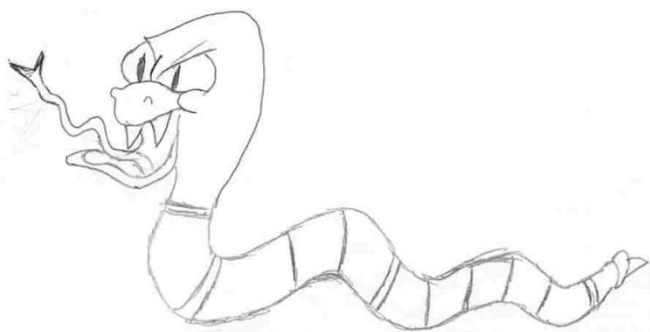
Castor Inácio



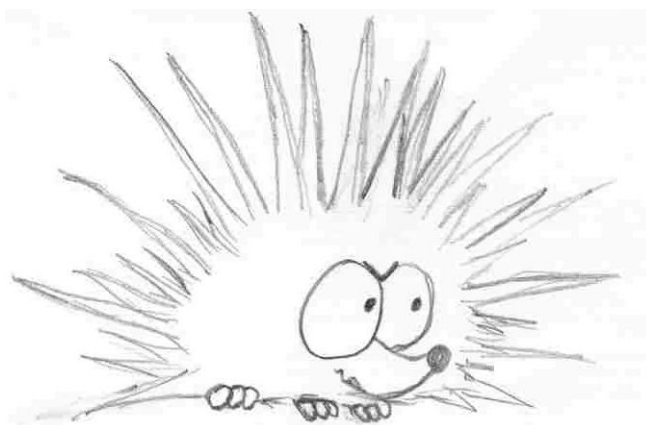
Splash Screen



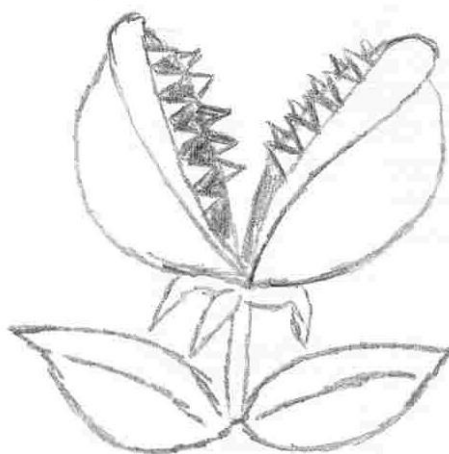
Abutre e Aranha



Cobra e Morcego



Ouriço e Piranha



Planta Carnívora

Sleepwalker



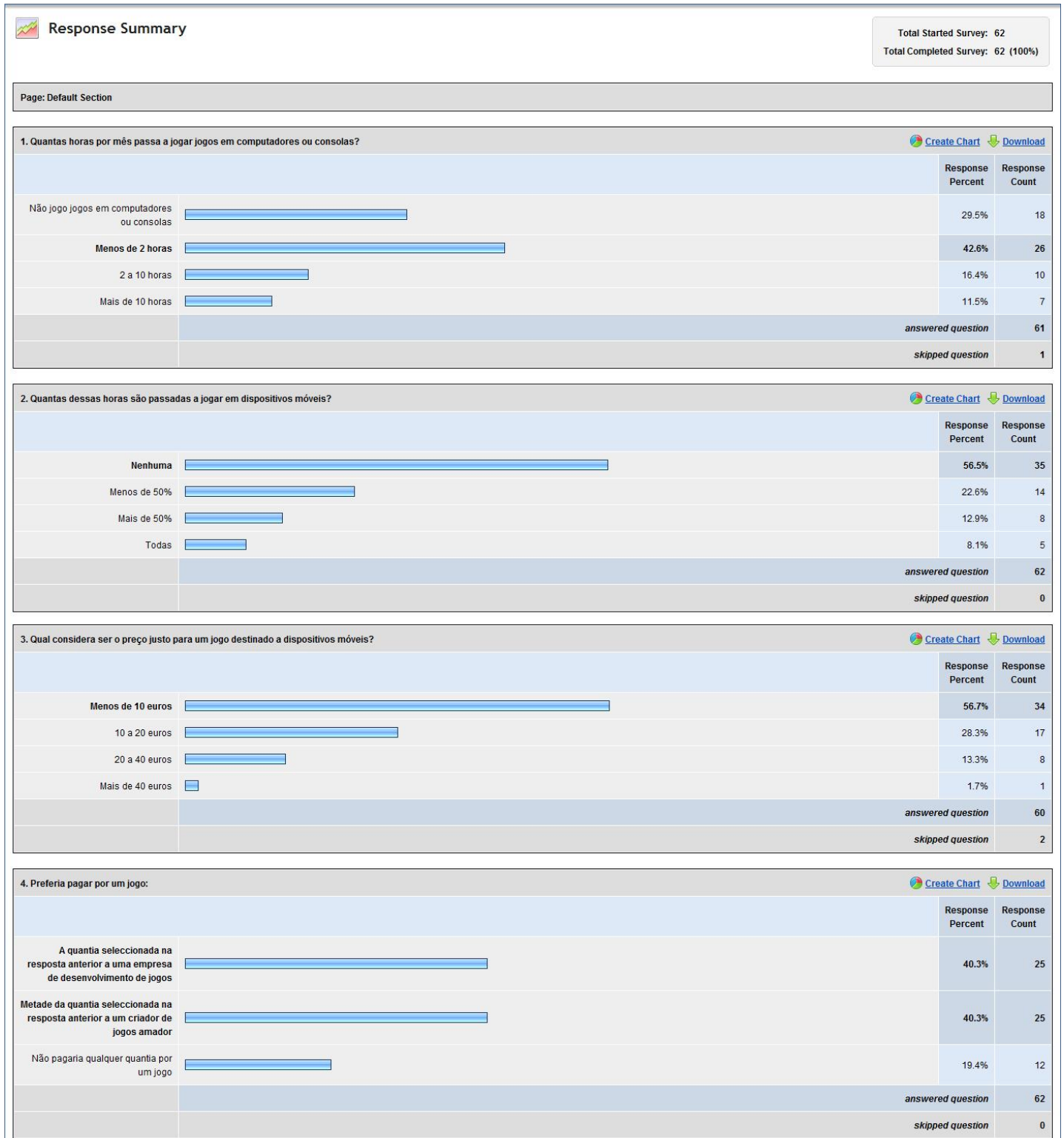
Godofredo Final

Anexo 4 – Folhas de *Sprites*



Anexo 5 – Inquéritos

Inquérito 1 – Análise de viabilidade do projecto



Sleepwalker

5. Quais são para si os elementos mais importantes num jogo?

[Create Chart](#) [Download](#)

	Response Percent	Response Count
Jogabilidade	91.8%	56
Aspecto visual do jogo	50.8%	31
Longevidade do jogo	24.6%	15
Preço baixo	29.5%	18
Outros	6.6%	4
answered question		61
skipped question		1

6. Quais os estilos de jogos que prefere?

[Create Chart](#) [Download](#)

	Response Percent	Response Count
Puzzle	44.3%	27
Plataformas	47.5%	29
Aventura grafica	26.2%	16
RPG - Role Playing Game	27.9%	17
Simulação	24.6%	15
Desporto	16.4%	10
Shoot'em up	9.8%	6
Beat'em up	1.6%	1
FPS - First Person Shooter	18.0%	11
Outros	24.6%	15
answered question		61
skipped question		1

Inquérito 2 – Aceitação

Response Summary

Total Started Survey: 62
Total Completed Survey: 62 (100%)

Page: Sleepwalker - As noites atribuladas de Godofredo

1. Em termos de originalidade, considera o jogo:

[Create Chart](#) [Download](#)

	Response Percent	Response Count
igual a muitos outros	1.6%	1
com alguns elementos originais, mas nada de novo	24.6%	15
com ideias novas suficientes para captar a atenção	60.7%	37
completamente original	13.1%	8
answered question		61
skipped question		1

2. Graficamente, o jogo é:

[Create Chart](#) [Download](#)

	Response Percent	Response Count
feio e mal ilustrado	0.0%	0
básico mas interessante	14.8%	9
com bons gráficos para o tipo de jogo que é	59.0%	36
muito bem concebido	26.2%	16
answered question		61
skipped question		1

3. Acha que o aspecto gráfico se relaciona com o tipo de jogo:

[Create Chart](#) [Download](#)


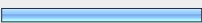
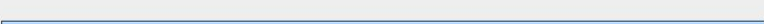
	Response Percent	Response Count
muito mal, um jogo deste tipo deveria ter uma imagem completamente diferente	0.0%	0
razoavelmente, mas ficava melhor com outro tipo de imagem	0.0%	0
bem, haveria melhores opções, mas a escolha funciona	32.3%	20
muito bem; é exactamente o tipo de gráficos adequado a este jogo	67.7%	42
answered question		62
skipped question		0




4. As animações do jogo:

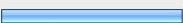

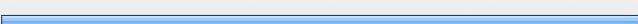

[Create Chart](#) [Download](#)

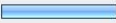
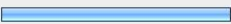

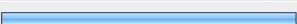
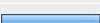
	Response Percent	Response Count
são básicas e rudimentares, parecem criadas por crianças	0.0%	0
funcionam mas podiam estar bastante melhores	4.8%	3
estão fluidas mas artificiais	9.7%	6
estão naturais mas "saltam" bastante	11.3%	7
estão fluidas e naturais	74.2%	46
answered question		62
skipped question		0

Sleepwalker

5. Os efeitos sonoros:		Create Chart	Download
		Response Percent	Response Count
são básicos e nada relacionados com o tipo de jogo		0.0%	0
estão engraçados mas não se adequam ao tipo de jogo		4.9%	3
são adequados ao tipo de jogo, mas estão muito fracos		19.7%	12
estão bem concebidos e trabalhados; soam bem e ligam bem com o jogo		75.4%	46
answered question			61
skipped question			1

6. Em termos de jogabilidade, o jogo:		Create Chart	Download
		Response Percent	Response Count
está muito fraco; nem deveria ser considerado um jogo		0.0%	0
joga-se para experimentar mas não cativa		3.2%	2
é interessante como ideia mas precisa de muito trabalho		32.3%	20
é bastante cativante e dá vontade de jogar mais		64.5%	40
answered question			62
skipped question			0

7. Em termos de dificuldade, o nível apresentado:		Create Chart	Download
		Response Percent	Response Count
é extremamente fácil e termina-se à primeira		0.0%	0
é fácil de resolver o puzzle mas exige alguma perícia		17.7%	11
é difícil chegar à solução mas a partir daí é fácil de completar		16.1%	10
é bastante difícil, ideal para ser um dos últimos níveis		62.9%	39
é praticamente impossível e não deveria fazer parte do jogo		3.2%	2
answered question			62
skipped question			0

8. Qual considera ser o preço justo para o produto final (4 mundos, 10 níveis por mundo) com base na experiência?		Create Chart	Download
		Response Percent	Response Count
Deve ser gratuito		11.3%	7
Menos de €4		22.6%	14
€4 a €8		27.4%	17
€8 a €10		29.0%	18
Mais de €10		9.7%	6
answered question			62
skipped question			0

Sleepwalker

9. Considera o jogo:		Create Chart	Download
		Response Percent	Response Count
totalmente amador, nota-se que foi feito em casa		0.0%	0
bem conseguido mas amador	<div><div></div></div>	3.2%	2
dentro do normal para um jogo cujo alvo são telemóveis e equipamentos móveis	<div><div></div></div>	50.0%	31
bem conseguido e equiparável a jogos para telemóveis de empresas especializadas	<div><div></div></div>	46.8%	29
answered question			62
skipped question			0

10. Relativamente à implementação, o nível apresentado:		Create Chart	Download
		Response Percent	Response Count
apresenta inúmeros bugs e/ou não funciona		0.0%	0
contém alguns bugs mas estes não têm grande influência na experiência		0.0%	0
apresenta apenas um ou outro bug	<div><div></div></div>	8.1%	5
está perfeitamente jogável sem qualquer problema	<div><div></div></div>	91.9%	57
answered question			62
skipped question			0